User Guide - English

FUJITSU

openFT V11.0 for Unix Systems

Managed File Transfer in the Open World

Edition August 2010

Comments… Suggestions… Corrections…

The User Documentation Department would like to know your opinion of this manual. Your feedback helps us optimize our documentation to suit your individual needs.

Feel free to send us your comments by e-mail to manuals@ts.fujitsu.com.

Certified documentation according to DIN EN ISO 9001:2008

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2008.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright and Trademarks

Copyright © Fujitsu Technology Solutions GmbH 2010.

All rights reserved.
Delivery subject to availability; right of technical modifications reserved.

All hardware and software names used are trademarks of their respective manufacturers.

This manual is printed on paper treated with chlorine-free bleach.

Contents

1 Preface 11

1.1 Brief description of the product 12
1.2 Target group 12
1.3 Concept of openFT for Unix systems manuals 13
1.4 Changes since the last version of the manual 15
1.5 Notational conventions 19
1.6 README files 19
1.7 Current information on the Internet 19
1.8 License provisions 20

2 openFT - the Managed File Transfer 21

2.1 Heterogeneous computer systems 24
2.1.1 File conversion 24
2.1.2 openFT product range 25
2.2 Heterogeneous networks 27
2.2.1 The OSI reference model 27
2.2.2 Position of the openFT product family in the OSI Reference Model 29
2.2.3 openFT partners 31
2.2.4 FTAM partners 32
2.2.5 FTP partners 33
2.3 Transferring files 35
2.3.1 Specifying the transfer start time 36
2.3.2 Controlling the duration of a request 37
2.3.3 Request queue 37
2.3.4 Automatic restart 39
2.4 File management 40
2.5 Remote command execution 41

2.6 Automation . . . 42
2.6.1 File transfer with preprocessing, postprocessing and follow-up processing . . . 42
2.6.1.1 Preprocessing . . . 43
2.6.1.2 Postprocessing . . . 44
2.6.1.3 Follow-up processing . . . 44
2.6.2 Program interfaces . . . 46
2.6.3 openFT script interface . . . 46
2.6.4 Job variables in BS2000 . . . 47
2.7 Further processing of openFT data . . . 48
2.8 Secure operation . . . 49
2.8.1 The FTAC function . . . 49
2.8.1.1 Features of the FTAC function . . . 49
2.8.1.2 Admission set . . . 51
2.8.1.3 FT profile (admission profile) . . . 52
2.8.1.4 Effects of an admission profile . . . 55
2.8.1.5 FTAC administrator . . . 57
2.8.2 Encryption for file transfer requests . . . 57
2.8.3 Logging openFT operations - the logging function . . . 59
2.8.4 Authentication . . . 61
2.9 Using openFT in a cluster . . . 63
2.10 Switching language interfaces . . . 64

3 File transfer and file management . . . 65

3.1 File names . . . 66
3.1.1 Unique file names for receive files . . . 66
3.1.2 BS2000/OSD file names . . . 67
3.1.3 File names in Unix systems . . . 70
3.1.4 Windows file names . . . 70
3.1.5 z/OS file names . . . 71
3.2 File passwords . . . 74
3.3 File types . . . 75
3.3.1 BS2000/OSD files . . . 75
3.3.2 z/OS files . . . 76
3.3.3 Unix and Windows files . . . 77
3.3.4 FTAM files . . . 79
3.3.5 Transfer of various file types . . . 80
3.3.6 Migrated files . . . 84

3.4 Transferring 7-bit, 8-bit and Unicode files 85
3.4.1 Code tables and coded character sets (CCS) 85
3.4.2 Specifying the CCS on a transfer request 86
3.4.3 Data conversion 87

3.5 Entries for the remote system 90
3.5.1 Defining the partner computer 90
3.5.2 Transfer admission 94

3.6 Options for file transfer 97
3.6.1 Maximum record lengths 97
3.6.2 Syntax rules 97
3.6.3 Compressed file transfer 98
3.6.4 Encrypted file transfer 99
3.6.5 Notifying results 99
3.6.6 Access mode 100
3.6.7 Preprocessing and postprocessing 100
3.6.8 Follow-up processing 103

3.7 File management 105
3.7.1 File management in the remote system 105
3.7.2 File management in the local system 106

3.8 Special points for file transfer with FTAM partners 108
3.8.1 Virtual filestore 108
3.8.2 Mapping file access rights 111
3.8.2.1 Outbound requests 111
3.8.2.2 Inbound requests 112
3.8.3 Mapping FTAM attributes to the real file system 114
3.8.3.1 Inbound mapping of FTAM attributes 115
3.8.3.2 Inbound mapping the document type 117
3.8.3.3 Access protection 119
3.8.3.4 Outbound mapping of the document type 120
3.8.4 FTAM diagnostic codes as per ISO 8571-3 123
3.8.5 Addressing via Application Entity Title (AET) 129

4 Working with openFT 133

4.1 The openFT Explorer for X Window 133

4.2 The openFT-Script interface 137

4.3 The openFT commands 138

4.4 Program interface 138

5 openFT commands for the user 139

5.1 Overview of the commands 139
5.2 Notational conventions 142
5.3 Output in CSV format 145
5.4 ft - Asynchronous file transfer 147
5.5 ftcanr - Cancel asynchronous requests 168
5.6 ftcredir - Create remote directories 171
5.7 ftcrep - Create an FT profile 174
5.8 ftdel - Delete a file in a remote system 190
5.9 ftdeldir - Delete remote directories 193
5.10 ftdelp - Delete FT profiles 196
5.11 ftedit - Load local or remote files in the openFT editor . . . 198
5.12 ftexec - Execute operating system commands in remote system . 200
5.12.1 Messages from the ftexec command 204
5.13 fthelp - Display information on the log record reason codes 207
5.14 ftinfo - Output information on the openFT system 208
5.15 ftmod - Modify file attributes in a remote system 210
5.16 ftmoda - Modify admission sets 217
5.17 ftmoddir - Modify attributes of remote directories 221
5.18 ftmodf - Modify the FTAM attributes of a local file 225
5.19 ftmodp - Modify FT profiles 231
5.20 ftmodr - Change the property of requests 249
5.21 ftmonitor - Call the openFT Monitor for displaying measurement data 251
5.22 ftmsg - Output a message box on a graphical display . . . 254
5.23 ftseti - Set an instance 255
5.24 ftshw - Display the attributes of one or more remote files . 256
5.24.1 Description of file attribute display 260
5.25 ftshwa - Display admission sets 265

5.26 ftshwf - Display the attributes of a local file 268
5.27 ftshwi - Display information on instances 270
5.28 ftshwl - Display log records 272
5.28.1 Description of log record output 279
5.28.1.1 Logging requests with preprocessing/postprocessing 279
5.28.1.2 Short output format of a FT or FTAC log records 279
5.28.1.3 Long output format of an FT log record 282
5.28.1.4 Long output format of an FTAC log record 286
5.28.2 Reason codes of the logging function 288
5.29 ftshwm - Display monitoring values of openFT operation . . 290
5.29.1 Description of the monitoring values 292
5.30 ftshwo - Display operating parameters 300
5.31 ftshwp - Display FT profiles 306
5.32 ftshwptn - Display partner properties 311
5.33 ftshwr - Display request properties and status 318
5.33.1 Output of the ftshwr command 321
5.33.1.1 Standard ftshwr output 321
5.33.1.2 Totaled ftshwr output 323
5.33.1.3 Detailed output from ftshwr 323
5.34 ncopy - Synchronous file transfer 331

6 openFT-Script Commands 355
6.1 Overview of the openFT-Script commands 355
6.2 ftcans - Cancelling an openFT-Script request 356
6.3 ftdels - Deleting an openFT-Script request 358
6.4 ftscript - Starting an openFT-Script request 360
6.5 ftshwact - Displaying the activities associated with an openFT-Script request 362
Description of the output 364
6.6 ftshws - Display openFT-Script requests 369

7 Program interface 373
7.1 Programming with C 373
7.2 Programming with JAVA 374

8 What if ... 375

8.1 Actions in the event of an error 375

8.2 Locked transfer admissions - possible causes and remedies 376

9 Messages 377

9.1 openFT messages 378
9.1.1 Messages applying to all commands 378
9.1.2 Messages applying to file transfer, file management and remote administration commands 379
9.1.3 Messages for administration commands and measurement data recording 408

9.2 FTAC messages 415

10 Appendix 419

10.1 Tool Command Library 419
10.1.1 ft_tar 420
10.1.2 ft_gzip 421
10.1.3 ft_b2u and ft_u2b 422
10.1.4 ft_mget - Fetching multiple files 423

10.2 Sample files 429

10.3 Structure of CSV Outputs 431
10.3.1 ftshw 431
10.3.2 ftshwa 433
10.3.3 ftshwl 435
10.3.4 ftshwm 438
10.3.5 ftshwo 442
10.3.6 ftshwp 445
10.3.7 ftshwptn 448
10.3.8 ftshwr 449

10.4 BSFT commands 453
10.4.1 BSFT session 453
10.4.2 Call BSFT 455
10.4.3 Functional overview of BSFT commands 457
10.4.4 Notational conventions 459
10.4.5 Command descriptions in alphabetical order 460
10.4.6 Messages from BSFT 484

Glossary 491

Abbreviations 515

Related publications 519

Index 521

Contents

1 Preface

The openFT product range transfers and manages files

– automatically,
– securely, and
– cost-effectively.

The reliable and user-friendly transfer of files is an important function in a high-performance computer network. The corporate topologies consist of networked PC workstations, which are additionally linked to a mainframe or Unix based server or Windows server. This allows much of the processing power to be provided directly at the workstation, while file transfer moves the data to the mainframe for further processing there as required. In such landscapes, the locations of the individual systems may be quite far apart. Fujitsu Technology Solutions offers an extensive range of file transfer products - the openFT product range - for the following system platforms:

- BS2000/OSD®
- Solaris™(SPARC®/Intel™), LINUX®, AIX®, HP-UX®
- Microsoft® Windows XP™, Windows Server 2003™, Windows Vista™, Windows™ 7 and Windows Server 2008™
- OS/390 and z/OS (IBM®) respectively.

1.1 Brief description of the product

openFT for Unix systems is the file transfer product for systems with a Unix based operating system.

All openFT products communicate with each other using the openFT protocol (previously known as the: FTNEA) as laid down by Fujitsu. Since a number of FT products from other software vendors also support these protocols, many interconnection options are available.

When used in combination with openFT-FTAM, openFT also supports the FTAM file transfer protocol (File Transfer Access and Management) standardized by ISO (International Organization for Standardization). This makes it possible to interconnect with even more systems from other vendors whose file transfer products support the same standard.

When used in combination with openFT-FTP, openFT also supports the FTP protocol. This makes it possible to interconnect with other FTP servers.

With the integrated FTAC function, openFT offers extended admission and access protection (FTAC stands for **F**ile **T**ransfer **A**ccess **C**ontrol).

1.2 Target group

This manual is aimed at users who wish to transfer or manage files using openFT for Unix systems. It explains how to use the FTAC function properly.

To understand this manual, it is useful to have a knowledge of the Unix based operating systems.

The manual covers Sun Solaris systems as well as portings to other Unix platforms. The operating system-dependent differences are described in detail in the Release Notices supplied on the respective product CD.

1.3 Concept of openFT for Unix systems manuals

The complete description of openFT and its optional components comprises four manuals. The description is divided among the manuals as follows:

- openFT for Unix systems - Installation and Administration

  The system administrator manual is intended for FT, FTAC and ADM administrators. It describes:
  - the installation of openFT and its optional components
  - the operation, control and monitoring of the FT system and the FTAC environment
  - the administration commands for FT and FTAC administrators
  - the configuration and operation of a remote administration server and a ADM trap server
  - important CMX commands.

- openFT for Unix systems - Managed File Transfer in the Open World

  The user manual is intended for the openFT user and describes:
  - the basic functions of the openFT product family,
  - the conventions for file transfers to computers running different operating systems,
  - details on implementing FTAM,
  - the openFT user commands,
  - the openFT-Script commands,
  - the BSFT interface,
  - the messages of the different components.

- openFT for Unix systems and Windows systems - C Program Interface

  This manual is intended for C programmers and describes the C program interface on Unix systems and Windows systems.

- openFT for Unix systems and Windows systems - openFT-Script Interface

  This manual is intended for XML programmers and describes:
  - the openFT-Script commands
  - the XML statements for the openFT-Script interface

Many of the functions described in the manuals are also available in the openFT graphical interface, the openFT Explorer. A detailed online help system that describes the operation of all the dialogs is supplied together with the openFT Explorer. The online help system also contains a complete description of the openFT commands.

1.4 Changes since the last version of the manual

This section describes the changes in openFT V11.0 for Unix systems compared to openFT V10.0 for Unix systems.

Remote administration

openFT instances from different platforms can be administered using a remote administration server that can run on a Unix or Windows system. Remote administrators are defined for this purpose. These remote administrators can enter the administration requests on the remote administration server or on another openFT instance. In the second case, they must specify a corresponding FTAC transfer admission on the remote administration server.

The attributes of the remote administrators and the address and access data of the openFT instances to be administered are defined in a central configuration file on the remote administration server. Access to the openFT instances to be administered is controlled by special admission profiles that are set up on the instances.

To this end, the following commands have been introduced or enhanced:

- New command *ftadm* for administering the openFT instances.
- New commands *ftimpc* and *ftexpc* for importing and exporting the configuration data on the remote administration server.
- New command *ftshwc* that allows remote administrators to view the instances they are allowed to administer.
- The *ftmodo* command has been expanded to allow the properties of the remote administration server to be defined.
- The *ftcrep* and *ftmodp* commands have been expanded to allow the admission profiles for remote administration to be defined.
- The *ftshwl* command has been expanded to make it possible to select on the basis of the new administration log records and their properties.

The openFT Explorer also makes these functions available and has been expanded accordingly.

ADM traps

ADM traps (= event-driven short messages) can also be sent to an ADM trap server if certain events occur during openFT operation. A Unix or Windows system configured as a remote administration server can act as the ADM trap server. The FT administrator of the ADM trap server can read the ADM traps. If remote administrators are defined on the ADM trap server for the openFT instance sending the trap, these remote administrators can also view the associated ADM traps.

To this end, the following commands have been introduced or enhanced:

- New command *ftshwatp* for viewing the ADM traps.
- The *ftcrep* and *ftmodp* commands have been expanded to allow the admission profiles for receiving traps on the ADM trap server to be defined.
- The *ftmodo* command has been expanded to allow the destination and scope of the ADM traps to be sent to be defined.

The openFT Explorer also makes these functions available and has been expanded accordingly.

Monitoring of openFT operation

Important monitoring data for openFT operation can be collected and output. This includes various values relating to throughput or processing time and current state indicators.

The administrator configures, activates and deactivates monitoring (using the *ftmodo* command with the *-mon*, *-monp* and *-monr* options or using the openFT Explorer).

The monitoring data can be output either with the new command *ftshwm* or using the openFT Monitor. The openFT Monitor can be started by entering the command *ftmonitor* or via the openFT Explorer. Administrator permissions are not required for displaying the monitoring data. The FT administrator must first activate monitoring with *ftmodo -mon=n*.

Monitoring data from partner systems can also be collected and output on the local computer using the openFT Monitor. In addition, the new preprocessing command *FTMONITOR allows special admission profiles to be defined that only permit monitoring. The monitoring data can be transferred to the local computer with *ncopy* or *ft* and saved in a file for subsequent evaluation, for instance. The openFT Monitor and the preprocessing command *FTMONITOR use the openFT protocol.

Extended security functions

- Secure FTP over SSL:
  Encryption is now also possible for outbound requests to a standard secure FTP server.
- 2048-bit RSA key:
  2048-bit RSA keys are now also supported.
- 256-bit AES key:
  256-bit AES keys are now also supported.

Prioritization of partners

Partners can be prioritized in the partner list. Requests to different partners that have the same request priority are processed in the order given by the partner priority entered.

File transfer and request queue

- The format of the target file can be specified in the transfer request (new option *-tff=* for *ft* and *ncopy*).
- New script *ft_mget* for synchronously fetching several files.
- Requests with the status SUSPEND are output when the request queue is displayed. It is also possible to select requests on the basis of the SUSPEND status when displaying requests (*ftshwr -st=s*).
- More entries have been reserved in the request queue for inbound requests than in previous versions in order to guarantee as far as possible that inbound requests do not need to be rejected because there is no more space in the request queue.

Logging and traces

- It is also possible to select on the basis of the request number and whether the request was for FTP when outputting log records (*-ff=l* and *-tid=* options for *ftshwl*).
- The logging ID can be up to 12 characters long.
- The traces have been optimized and the structure and names of the trace files have been changed.

Other changes

- Additional code tables are supplied.
- With *ftexec*, it is also possible to enter the commands to be executed via *stdin*.
- A default admission profile can be set up for each user ID (*@s* option for *ftcrep* and *ftmodp*). This is primarily intended for partners who must specify the transfer admission in a specific form.
- Admission profiles now have a timestamp with the most recent change date. This is output with *ftshwp -l*, for instance.
- A uniform method of activating and deactivating the asynchronous inbound server (openFT, FTAM, FTP, ADM) has been introduced (new option *-acta=* for *ftmodo*).

1.5 Notational conventions

The following notational conventions are used throughout this manual:

`typewriter font`
: typewriter font is used to identify entries and examples.

italics
: In running text, names, variables and values are indicated by italic letters, e.g. file names, instance names, menus, commands and command options.

: indicates notes

Additional conventions are used for the command descriptions, see page 142.

1.6 README files

Information on any functional changes and additions to the current product version can be found in product-specific README files.

1.7 Current information on the Internet

Current information on the openFT family of products can be found in the internet under *http://ts.fujitsu.com/openft*.

1.8 License provisions

License provisions apply to the use of *libxml2* and Secure FTP. You can find details in the manual "openFT V11.0 for Unix Systems - Installation and Administration".

2 openFT - the Managed File Transfer

Managed File Transfer is a term that documents the high performance of openFT products. Such high demands on corporate file transfer result, on the one hand, from the variety of hardware and software commonly installed today and, on the other, from the different needs your company has with respect to file transfer itself. A further important aspect of enterprise file transfer is provided by the options for automation and the security functions offered by openFT. In addition, central administration of an openFT network and presentation of the operating states make openFT a managed file transfer system.

Fujitsu Technology Solutions offers a comprehensive openFT product range for Managed File Transfer, which can be used to operate **heterogeneous computer systems** (hardware and software) of many manufacturers ranging from mainframe systems to the PC. openFT products can be used in various operating systems such as Windows, Unix systems, BS2000/OSD, z/OS and others.

Even **heterogeneous networks** such as TCP/IP, NEA, ISO, X.21/X.25, ISDN and GSM mobile telephony or MODACOM pose no problem for openFT. The continual integration of new platforms and network types guarantees high availability of the openFT products, also in the future. Not all networks are supported on all platforms.

The integration of the **ISO 8571 FTAM standard** (File Transfer, Access and Management) guarantees uniform interfaces for requests to openFT partners and any FTAM partners (not available under z/OS).

Support for the **FTP protocol** makes it possible to connect to FTP servers and FTP clients on any required platform.

Functions such as request storage, automatic restart, job and file management, follow-up processing, resource management, program interfaces, encryption and authentication indicate the wide range of services offered by openFT products, thus making them truly suitable for Managed FIle Transfer.

Request storage makes it possible to start **asynchronous file transfer** at any desired time, e.g., to save charges or to wait for the occurrence of specific events. The **automatic restart** feature ensures a consistent continuation of file transfer after the correction of a fault, e.g., a network or processor failure.

Automation is achieved, among other things, via facilities for preprocessing and follow-up processing:

- Local or remote **preprocessing** enables data to be created within a send or receive request by starting a job, for example, and then transferring it then to the local or remote system.
- Local or remote **postprocessing** enables the data transferred to be processed further within a send or receive request.
- Preprocessing as well as postprocessing can be executed within a request.
- **Follow-up processing** permits any job to be started just after file transfer. You can make the start of follow-up processing dependent on the success of the file transfer.

The **program interfaces** permit the implementation of openFT functions in programs.

File management in the remote and local systems provides facilities for modifying file attributes. for example.

The **resource control** allows you to store file transfer requests at any time and have them issued automatically when the partner system is available. Connection to the BS2000 job variable mechanism is also possible.

With **synchronous file transfer**, you can use the openFT functions for file transfer interactively, e.g. directly on your screen.

Protection of the data inventory is becoming a priority issue in companies in view of the open nature of today's networks. The **FTAC functionality** (optional in openFT for BS2000/OSD and openFT for z/OS) integrated in openFT products offers comprehensive and individually scalable protection functions:

– decoupling of transfer admissions and login admission
– access rights dependent on the partner systems
– user-specific access rights
– flexible access right levels
– recording of every authorization check

The **logging** of data transfer requests and authorization checks permits evaluation of previous request and access, thus providing a further security feature.

The **encryption** of request description and transfer data is another protection level provided by openFT. Request description data include the authorization data for the transfer of and access to data (e.g. transfer admission, file password). In addition, it is possible to connect to system security functions such as SECOS on BS2000, RACF and ACF2 on z/OS.

In openFT (version 8.1 and later) expanded identity checking (i.e. **authentication**) of the communications partner is offered for requests using openFT partners. It is based on addressing network-wide, unique IDs for openFT instances and the exchange of partner-specific key information.

2.1 Heterogeneous computer systems

One strength of the openFT products is their capability for linking different computers, particularly computers from different manufacturers running various operating systems. The precondition for file transfer between two computers is that a transport connection exists between these two computers and that one of the openFT products, an FTAM product or an FTP application is installed on the computers.

The openFT products are matched for optimum interoperability. They retain file structures and attributes during file transfer. openFT products cannot override the conventions that apply to the operating system. Data conversion may be necessary to ensure that characters are represented correctly when performing transfers between certain operating systems.

2.1.1 File conversion

The coding, i.e. the system-internal representation of individual characters, letters and digits, depends on the operating system. The data must then be converted because

– Unix systems and Windows computers use an ISO-8859-x-Code internally. This code is described in ISO Standard 8859. The ASCII Code (American Standard Code for Information Interchange) is a subset of ISO-8859-1.
– BS2000/OSD systems and z/OS computers, on the other hand, normally use an EBCDIC (Extended Binary-Coded Decimal Interchange Code).

Data conversion between openFT partners always applies to the characters with which parameter values (e.g. file names, user IDs, follow-up processing strings, etc.) are transferred.

The conversion of file contents, by contrast, is only relevant for files to be transferred in text format; no data conversion is performed by openFT when transferring files in other formats (binary, transparent, etc.).

Please note that the openFT partner codes use the same character repertoire. If this is not the case, some of the characters in the text file (e.g. umlauts) may not be represented correctly. If you transfer files with openFT partners as of V10, you can assign the "Coded Character Sets" that are to be used for local and remote data conversion in the request. It is also possible to transfer Unicode files with these partner systems, see section “Transferring 7-bit, 8-bit and Unicode files” on page 85.

2.1.2 openFT product range

The tables below provide an overview of the openFT product range, showing the openFT products currently available for your computer.

openFT product range

Product	Operating system	Comment
openFT for Unix systems	AIX, LINUX, HP-UX, Sun Solaris	Additional systems on request
openFT for BS2000/OSD	BS2000/OSD	BS2000 systems from Fujitsu Technology Solutions
openFT for Windows systems	Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7	Intel architecture
openFT for z/OS	z/OS	z/OS systems from IBM

openFT add-on products

Product	Operating System	Comment
openFT-FTAM for Unix systems	AIX, Linux, HP-UX, Sun Solaris	Unix systems
openFT-FTAM for Windows systems	Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7	Intel architecture
openFT-FTAM for BS2000/OSD	BS2000/OSD	FTAM functionality for BS2000 systems from Fujitsu Technology Solutions
openFT-FTP for Unix systems	AIX, Linux, HP-UX, Sun Solaris	Unix systems
openFT-FTP for Windows systems	Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7	Intel architecture
openFT-FTP for BS200/OSD	BS200/OSD	FTP functionality for BS2000 systems
openFT-FTP for z/OS	z/OS	FTP functionality for z/OS systems
openFT-AC for BS2000/OSD	BS2000/OSD	FTAC functionality for BS2000 systems
openFT-AC for z/OS	z/OS	FTAC functionality for z/OS systems
openFT-CR	All platforms of the openFT product family	Data encryption (restricted to export)

2.2 Heterogeneous networks

A group of interlinked computers and other devices is referred to as a network. When computers with the same type of communications structure are linked, we use the term homogeneous network.

The term heterogeneous network is used to denote a computer network in which computers intercommunicate with different communication architectures. Essential properties of computer networks are distances to be covered, the type transmission route, the utilization of public services and the type of protocols, i.e. the entire range of rules and regulations which must be observed for information transfer.

The most renowned networks supported by openFT are TCP/IP, NEA, ISO, SNA, X.21/X.25, ISDN, GSM mobile telephony or MODACOM. Not all network types are supported on all platforms.

Network management in heterogeneous networks are based on **SNMP** (Simple Network Management Protocol) in most cases.

The openFT products support the SNMP-based network management and thus underline their import in open networks.

2.2.1 The OSI reference model

In order to exchange data, systems must be able to intercommunicate. Communication is possible only if the computers involved use the same file formats for data exchange and observe an agreed behavior during transfer. The sum of the conventions and file formats for communication is referred to as a protocol. Protocols are defined by the manufacturer (for example openFT protocols) on the one hand, and on the other by committees which define manufacturer-independent protocols. ISO (International Organization for Standardization) provides the OSI Reference Model (**O**pen **S**ystems **I**nterconnection), the best-known model for communications architecture and the most comprehensive collection of protocols.

The OSI Reference Model structures the communications functions of computer systems and provides a foundation for standardization of protocols and services. It specifies which functions the components involved in communication must provide.

The OSI Reference Model consists of seven hierarchically structured layers. Each layer is assigned specific communication functions.

Layers	Designation	Functions	
Layer 7	Application Layer	Coordinates and controls the performance of communication tasks for an application	**A**PPLICATION
Layer 6	Presentation Layer	Regulates the form of information presentation and thus permits user/device-independent communication	
Layer 5	Session Layer	Regulates the sequence of communication	
Layer 4	Transport Layer	Regulates the reliable exchange of data between two communications partners	**T**RANSPORT
Layer 3	Network Layer	Regulates the exchange of data between two terminal systems (computers)	
Layer 2	Data Link Layer	Secures the transmission on individual subroutes of the entire transmission route (procedures)	
Layer 1	Physical Layer	Provides the physical connection (via the medium used for transmission)	

OSI Reference Model

The individual layers use the service of the layer immediately below and provide a precisely defined service to the layer above. Only the physical layer must provide its service together with the physical medium. The active elements within a layer, which provide the functions, are referred to a instances.

Each layer is specified by the service it provides, and the services it uses from the layer below it. During communication, the various computers interoperate on the same layer, using common protocols.

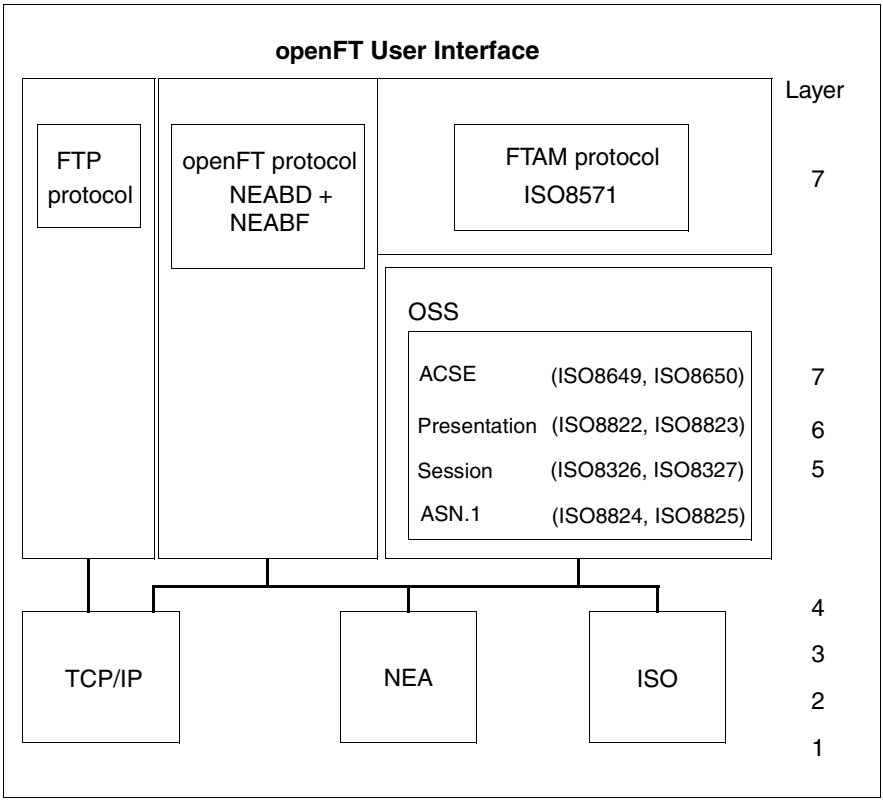
The functionality of each layer in the OSI Reference Model can be provided by various protocols as a rule. Decisive for the communication is that the direct partner instances use the same protocol for a particular task. For this purpose, profiles are defined.

A profile is understood as precise specification of which protocols or which protocol variants are to be used on which layer to perform a particular task. Profiles are stipulated by national or international organizations or communities.

2.2.2 Position of the openFT product family in the OSI Reference Model

The openFT products belong to the application layers (Layers 5 - 7) of the OSI Reference Model. They support the standardized openFT protocol and the FTAM protocol ISO8571 standardized by ISO and the File Transfer Protocol (FTP) defined by RFC959.

The openFT products can use a variety of different transport systems with different transport protocols. The following diagram shows the possible combinations of application and transport protocols for file transfer:



Protocols supported by openFT in the environment of the OSI Reference Model

For an overview of the transport systems and protocols that permit the operation of openFT products, please refer to the relevant product data sheets.

2.2.3 openFT partners

openFT can perform file transfer and file management between partner systems which support the openFT protocols NEABD and NEABF in the application layers.

These partner systems are referred to below as openFT partners. openFT partners can run on mainframe platforms (BS2000/OSD, z/OS) and on open platforms (Unix systems, Windows systems).

Depending on the particular transport system software, a variety of transport protocols may be used:

– TCP/IP transport protocols
– NEA transport protocols
– ISO transport protocols

The range of functions is largely identical for a given openFT version across the different platforms, and any minor differences are the result of the operating system used.

These protocols, which were originally referred to as FTNEA protocols, have been opened, so there are now also products from other manufacturers that support these protocols.

2.2.4 FTAM partners

The FTAM extension available in openFT also enables openFT to perform file transfer and file management with partner systems which support ISO protocols in layers 5 - 7 of the OSI Reference Model. In the rest of this manual, these systems are referred to as FTAM partners, since they use the protocols for file transfer defined in the international standard ISO 8571 (FTAM, File Transfer, Access and Management).

BS2000/OSD also require the OSS software package to implement layers 5 - 7.

Implementation of FTAM Standards in openFT

A subset of the complete functional scope of the base standards has been selected in accordance with international and European profiles ISO/EN ISP 10607-3 and ISO/EN 10607-6. This functional standardization has, in turn, been harmonized with other functional standards (and implementation agreements), e.g. the corresponding implementation agreements of IGOSS in North America and corresponding profiles in Asia and Australia.

ENV 41204 and ENV 41205 are the old, nevertheless still applicable, designations for EN 10607-3 and EN 10607-6 and their contents are identical to the international profiles ISO/IEC ISP 10607-3 (1990) and ISO/IEC ISP 10607-6 (1990) agreed by ISO. EN 10607-3 and EN 10607-6 contain additional European character repertoires.

These profiles specify the file attributes actually used, for example, and the operations permitted with these attributes, irrespective of the operating system used. A **virtual filestore** is used to permit presentation across several operating systems; here, the contents of the real store are transferred with a representation of the file attributes in accordance with the standard. Conversion of the file attributes to FTAM Standard in the operating system and vice versa is part of the FTAM functionality. There are three groups of file attributes: kernel group, storage group and security group (see page 108).

Compliance with the FTAM standard also restricts the functional scope offered by openFT protocols. Transfer of follow-up data to FTAM partners is not possible with the protocol.

The mapping mechanism between the real filestore and the virtual filestore is described in detail on page 108.

2.2.5 FTP partners

Alongside openFT and FTAM partners, it is also possible to address FTP servers.

If the FTP protocol is used then only communication via TCP/IP is possible. Furthermore, a number of special considerations apply when FTP servers are used compared to openFT partners. These are for the most part due to limitations in the FTP protocol:

- No restart is performed.
- Encryption is only possible for outbound requests to an FTP server that provides support for Secure FTP with the TLS protocol. This requires openFT-Crypt (openFT-CR delivery unit) to be installed.
- If encryption of the user data is required and the standard Secure FTP server does not provide encryption, the request is rejected. If encrypted transfer of the user data is required, the login data is also encrypted. If encryption of the user data is not required, the login data is only encrypted if the standard Secure FTP server provides this. No mutual authentication is carried out.
- Coded character sets are only supported locally; specifications for the partner system cannot be transported by the FTP protocol.
- When files with a record structure are transferred in binary format, the record structure is lost. The contents of the records are stored in the destination file as a byte stream.
- File attributes are not supported by the FTP protocol. This means that the modification date and maximum record length are not taken over for the destination file.
- If the *ftexec* command is issued to a mainframe over the FTP protocol, the *-t* option must be used. The *-b* option (default) is rejected in the remote system with a message indicating that the file structure is not supported.
- Follow-up processing is only possible on the local system or by specifying the FTAC profiles.
- The modification date cannot be taken over for the destination file. As a result, the modification date of the destination file is set to the transfer date. This is of particular importance when comparing file hierarchies.

- If an FTP server does not provide the information as to whether a symbolic link refers to a file or a directory when listing directories, the link is by default shown as a file in openFT Explorer (on Unix and Windows systems).
- The maximum record length of the send file is not passed to the receiving system. This has an impact when transferring files to a mainframe system such as BS2000/OSD or z/OS. In this case, the default maximum record length applies in the receiving system. If a record in the file exceeds this length, the request is cancelled with the message “File structure error” (return code 2210 in log record).
- The size of the send file is not passed to the receiving system. This has an impact when transferring files to a mainframe system such as BS2000/OSD or z/OS. The maximum file size is derived from the default value that is used by openFT for primary and secondary allocation and by the maximum number of file extents defined by the system, see section “BS2000/OSD files” on page 75 and section “z/OS files” on page 76. If a file exceeds this size, the request is cancelled with the message: “File gets no more space”.
- The 'do not overwrite' option (*-n*) can have a different effect because this option cannot be passed to the responder, and the initiator must check whether the file already exists in the partner system. This has the following consequences:
  - It is possible for a request with the ’do not overwrite’ option (*-n*) to overwrite a file that has been created by a third party in the period between the check being performed by the initiator and the actual transfer.
  - If 'overwrite' is specified in an FTAC profile (*-wm=o*), and if the file to be transferred does not yet exist, a request using this profile will still be executed, even if 'do not overwrite' (*-n*) is set in the request.
- If you access password-protected mainframe files with a standard FTP client, e.g. in text format (C'password') or hexadecimal format (X'0A6F73'), you must append the password to the name of the remote file separated by a comma.

  Example

  ```
  put localfile remotefile,X'0A6F73'
  ```

Please note that the other openFT functions (preprocessing and postprocessing, FTAC, etc.) can only be used if openFT is used as the FTP server on the system, where preprocessing and postprocessing are to be performed. Problems may also occur when addressing FTP servers which send an unexpected layout when listing directories.

2.3 Transferring files

The main function of openFT is to transfer files between two partner systems. To do this, you must issue a file transfer request in the local system. This request can be used either to send a file to a remote system or to fetch a file from a remote system to the local system. A partner system can also send files to your local system or fetch one from your local system.

Requests issued from you local system are referred to as **outbound requests** (sent from outside). Requests issued from the remote system are referred to as **inbound requests** (received from outside).

In a file transfer request, you can specify whether the file to be transferred is a text file or whether it contains unstructured or structured binary data. This determines the handling of the data during transmission; see the section “File conversion” on page 24. The so-called “transparent” file format plays a special role here: you can use this format to store BS2000 files with all their properties in the receive system without conversion. This is necessary, for example, when a Unix or Windows system is used to distributed BS2000 software.

Preprocessing, postprocessing and/or follow-up processing can be agreed for all file transfer requests to openFT partners. You may specify follow-up processing for successful and failed transfers both in the local system and in the remote system. For details of how to use the preprocessing. postprocessing and follow-up processing features, see the section “File transfer with preprocessing, postprocessing and follow-up processing” on page 42.

You should not process a file further until transfer is completed; otherwise, inconsistencies may result.

You may decide for yourself when openFT is to carry out your transfer request. Either immediately or at a particular time which you can specify. openFT always performs a synchronous request immediately. If a request is to be performed later, you must start an asynchronous request and specify the time of its execution.

Compressed transfer

When issuing a request, you may specify whether the file is to be transferred in a “compressed” form and the type of compression that is to be used (byte compression or zip compression).

Data compression can be used to:

– shorten transmission times
– reduce the load on the transmission paths and
– reduce data transmission costs.

2.3.1 Specifying the transfer start time

When you start a **synchronous request**, the file is transferred immediately. During the entire transmission period, a display on screen allows you to follow the progress of the file transfer and you have the advantage of knowing immediately whether or not the transfer was successful. You can use the result as decision criterion for further steps. If transfer failed because the partner was not available, for example, the file transfer is aborted and you must restart the request later.

In the case of an **asynchronous request**, openFT transfers the file either at the next possible time or at the time you specify. This allows the file transfer to be started at a time when the partner is available, or when transmission charges are particularly low. The request is stored in a request queue and you receive confirmation that the request has been accepted. Your system is thus immediately free for other tasks and you do not have to take care of executing the request. Thus, for example, if it is not possible to set up a connection for file transfer at a particular time, openFT re-attempts start of file transfer at defined intervals; even if a fault occurs during transfer, it is restarted automatically.

You can start several asynchronous requests. The requests are placed in a request queue until they are successfully executed, or cancelled by you or their maximum lifetime as set globally has been reached (see the section “Controlling the duration of a request” on page 37). You can use the request queue to obtain information on all request that have not yet been executed.

Requests issued by a remote system, i.e. inbound requests, are always executed as asynchronous requests in the local system by openFT.

2.3.2 Controlling the duration of a request

An asynchronous openFT request remains in the request queue until it is fully executed or explicitly deleted or until its lifetime, which can be set via an administration parameter, expires.

When issuing an asynchronous request, however, you may specify a time at which the request is to be deleted, or the file transfer is to be canceled (cancel timer). In this way, you can avoid tying up resources for partners who are temporarily unavailable, or when network problems are encountered.

2.3.3 Request queue

The request queue stores all asynchronous file transfer requests which have not yet been executed. You may display these on screen at any time. The information displayed will include:

– the transfer direction
– the operational status of the request
– the number of bytes already transferred
– the initiator of the request
– the local file name, for outbound requests also the remote file name.
– the partner system involved
– follow-up processing
– diagnostic information

The byte counter in the request queue is updated at regular intervals, so that you can keep up-to-date on the progress of file transfer.

You may delete requests change the order of the requests in the request queue (priority control).

For information on requests that have already been completed, use the logging function (see the section “Logging openFT operations - the logging function” on page 59.

Priority control

The requests are processed according to the FIFO principle (FIFO = First In First Out), i.e. the request issued first is processed first. Two priority classes (normal/low) are possible. You can control the processing of a request by:

– explicitly specifying the priority of a request
– changing the priority of a request in the request queue
– changing the queue of the request queue, i.e. placing requests at the start or end of a list of request with the same priority

Prioritization of partners

Partners can be prioritized in the partner list. This priority only applies to requests that have the same request priority, but are sent to partners with different partner priorities. Otherwise, the request priority overrides the partner priority.

The list below shows the sequence in which requests are processed if requests with different request and partner priorities are present.

Processing sequence	Request priority	Partner priority
1	normal	high
2	normal	normal
3	normal	low
4	low	high
5	low	normal
6	low	low

2.3.4 Automatic restart

In the event of file transfer being interrupted for any reason, openFT provides for secure restart. This means that network problems, for example, present no difficulty to openFT, since openFT automatically continues transfer as soon as it becomes possible again.

The storage of the request in the request queue and the so-called restart points for the basis for automatic restart. These are the security points with which the two partner systems are synchronized at regular intervals during file transfer. If transfer is interrupted, it is continued as soon as possible starting at the last security point. You can therefore rest assured that not one single bit is lost and nothing is added during file transfer.

The fixed timing between security points ensures that no unnecessary security points are set for fast lines, and that the intervals are not too long for slow lines.

2.4 File management

In addition to file transfer, openFT offers the option of managing files in the remote and local and remote systems. You can perform file-management actions both with openFT statements and as processing within a file transfer request. It is expedient, for example, to formulate the necessary conditions for transfer or follow-up processing in the remote system prior to start of file transfer. This can be useful when creating file management requests prior to file transfer to the remote system, or when setting up conditions for follow-up processing, for example.

Furthermore, local or remote systems can be controlled from a Windows or Unix system via a user-friendly interface similar to the Windows standard, without the user having to be acquainted with the syntax of the remote system.

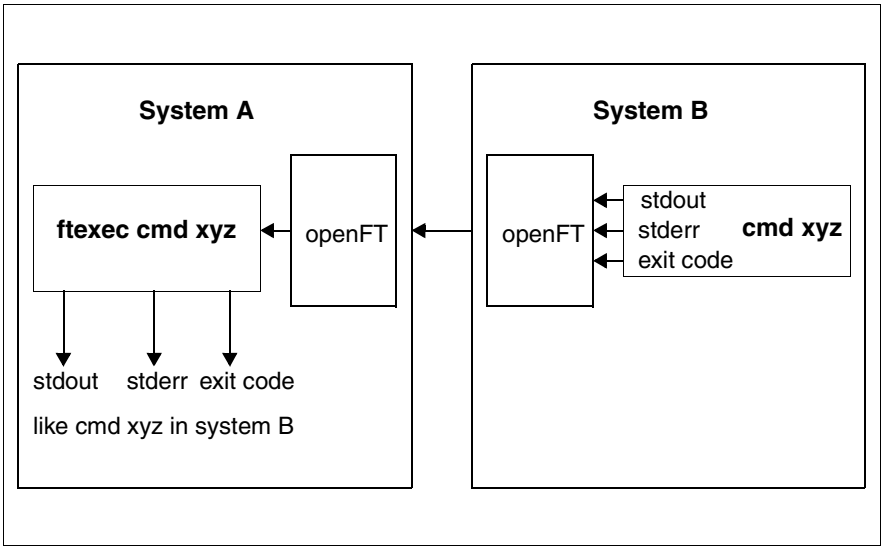
You can perform the following actions with via file management:

– rename files
– delete files
– query file attributes, e.g. the size of a file
– modify file attributes, e.g. access rights
– display directories
– create directories
– rename directories
– delete directories

2.5 Remote command execution

openFT for enables operating system commands to be executed on remote systems and can return the exit codes and outputs of such commands as if they were executed on the local system. This makes it possible to integrate remote commands transparently in local command procedures.

The following diagram clarifies the concept of remote command execution.



openFT concept for remote command execution

2.6 Automation

openFT provides job management functions such as file transfer with preprocessing, postprocessing and follow-up processing, connection to the job variable mechanism in BS2000, and the use of file-transfer functions in dialog procedures and via program interfaces.
Automation is also supported by the option for controlling the start time and lifetime of requests; see the corresponding sections. The creation of unique file names by openFT makes it easier to design applications and reduces the amount of updating work to be done.

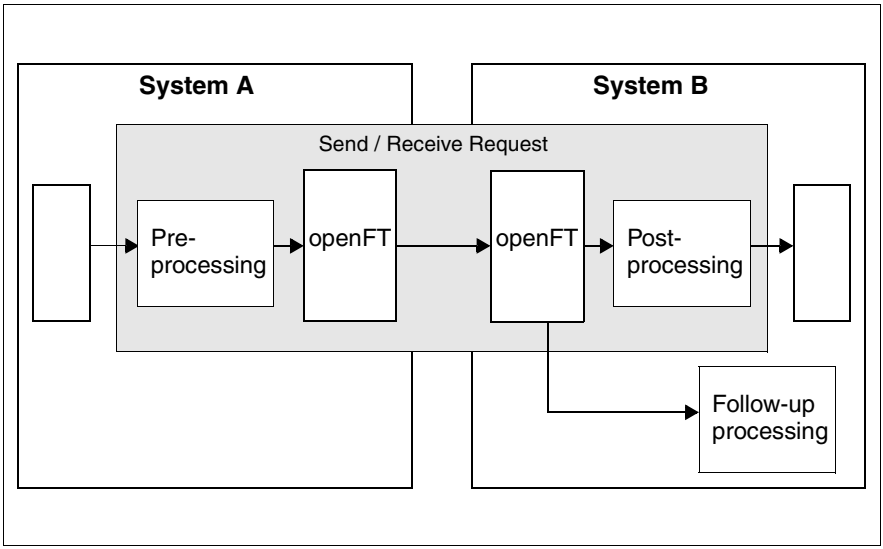
2.6.1 File transfer with preprocessing, postprocessing and follow-up processing

For a file transfer, you can specify

– whether any preprocessing or postprocessing is to be done within a request. Preprocessing in the sending system and postprocessing in the receiving system are always possible and can also be combined within a request.

– whether any follow-up processing is to be performed after the file transfer. Follow-up processing can be defined for successful and unsuccessful file transfers both for the local and the remote system.

The following diagram clarifies the concept of a file transfer with preprocessing, postprocessing and follow-up processing.



openFT concept for preprocessing, postprocessing and follow-up processing

Pre- and postprocessing always take place within the openFT request, and follow-up processing always take place after the request.

In order to prevent system resources from being unnecessarily tied-up in a continuous processing loop, requests should be provided with a specified abort time if necessary.

2.6.1.1 Preprocessing

During preprocessing, you can, within a file transfer request, prepare the send data **before** the transfer These could be operating system commands, program calls or procedure calls, in order to create or prepare the data before the transfer. The commands can, for example, extract information from a large data base (data base query), or prepare data (compress, encrypt), in order to subsequently pass it to openFT for file transfer.

2.6.1.2 Postprocessing

During postprocessing you can, within a file transfer request, process the received data using one or more commands **after** the actual transfer. To do this, you can execute commands, e.g. operating system commands, a program call or a procedure call. The command(s) can, for example, decode/uncompress data which has been encrypted or compressed using external routers.

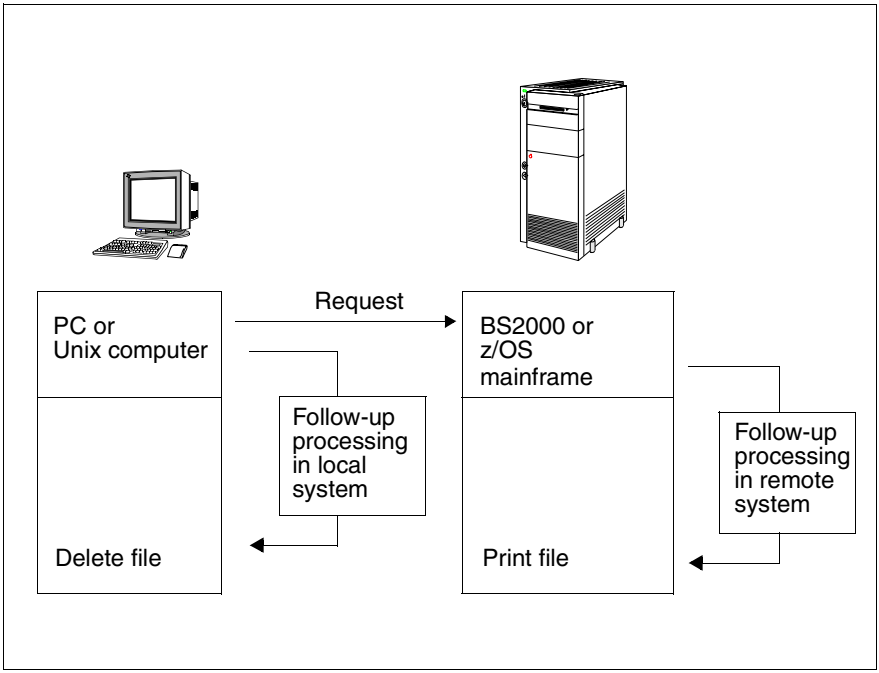
openFT requests with remote preprocessing or postprocessing can also be transferred by older versions of openFT or FT. It is important that a version of openFT that supports postprocessing is used in the remote system.

2.6.1.3 Follow-up processing

The "follow-up processing" option which is available in openFT enables you to execute sequences of statements or commands in the local and/or remote system depending on the positive or negative result of file transfer. If you specify follow-up processing for the remote system, you must observe the syntax of the operating system used on the remote system. When using commands, openFT provides variables which are replaced by the values in the file transfer request when the commands are executed.

Example

In the headquarters of a supermarket chain, there is a mainframe computer running BS2000 or z/OS. The branch office has Windows or Unix workstations. Every Saturday, the branch manager issues a request to transfer the file that contains a prepared list of the weekly sales. This file is transferred to the processor at the headquarters using openFT. The follow-up processing for the transfer request specifies that the file should be printed on the mainframe and then deleted from the branch computer if file transfer is successful.



File transfer with follow-up processing

2.6.2 Program interfaces

The program interface in openFT offers extensive automation capabilities. You can, for example, automate the issue of requests and request management in openFT, create your own user interfaces for openFT or integrate file transfer functions in other applications. In addition to the Java and C interface, an OCX interface is provided for Windows systems.

2.6.3 openFT script interface

openFT-Script provides a script language in XML notation which comprises the following openFT functions which are familiar to users from the command or C interface:

– Asynchronous file transfer

– Create directories in the remote system

– Delete files or directories in the remote system

– List directories in the remote system

– Run command scripts in the remote system

All openFT-Script functions can also be applied to local files or directories.

In addition, openFT-Script possesses the following advantages compared to the above-mentioned interfaces:

– Logically interdependent individual requests can be combined in a single request thus permitting simple monitoring.

– Individual requests can be run in sequence or in parallel.

– openFT-Script can restart. If an openFT-Script request is interrupted at a specific individual request then the openFT-Script request is resumed at this point on restart.

– openFT-Script requests can be monitored and interrupted in the openFT Explorer via the *Ftscript Requests* object directory.

– Alternative actions can be defined if errors occur (e.g. partner not accessible, file not present etc.).

2.6.4 Job variables in BS2000

openFT for BS2000/OSD offers the option of using a job variable to monitor a file transfer request. The name of the job variable, as well as any password required to access this variable are specified with the file transfer request. The job variable that monitors a request is also used to identify the request and may be used as a selection criterion to abort the request or obtain information about it. Further, it can be used for event control purposes, or to change request priorities.

2.7 Further processing of openFT data

In order to permit openFT data (*ftshwl*, *ftshwo*, etc.) to be processed further by external procedures, openFT offers the so-called CSV (**C**omma **S**eparated **V**alue) output format. In this format, each block of information is output to one line of text, with the individual items of information in an "output record" being separated by semicolons. The first line is a header and contains the names of the items of information, also separated by semicolons.

Such output could then be processed further by programs which support CSV formats (e.g. Microsoft Excel™ under Windows) and could hence be used, among other things, to easily implement an accounting system for the used resources (e.g. transfer requests).

2.8 Secure operation

Open networks, security during file transfer and data management are terms that need not be contradictory. openFT offers the following functions for secure operation are:

– individual settings for transfer and access rights with the FTAC function
– data encryption during the transfer
– logging function that can be enabled/disabled
– automatic encryption of the request description data
– Checking the communication partner using authentication

You can use these functions to make your system safe.

2.8.1 The FTAC function

With the FTAC function of openFT, you have all the options in your hand to make your system as secure as possible and as safe as it needs to be. FTAC stands for “File Transfer Access Control”.

FTAC offers the following protection mechanisms for your system:

– decoupling of FT transfer and login admissions
– access rights dependent on the partner systems
– user-specific access rights
– flexible access right levels
– recording of every authorization check
– simple application

2.8.1.1 Features of the FTAC function

For file transfer, a distinction is made between various functions. For access protection, the file transfer function being executed by the system is decisive. At first glance, there are only two such functions:

– sending a file and
– receiving a file.

Sending a file entails transmitting data from the system to be protected, while receiving a file involves the transfer of data into this system. However, for reasons of data security it is also important to know who requested a function in the system being protected. In FT terminology, this person is referred to as the initiator or submitter of the FT request. Initiators can be divided into two groups:

– those in the system being protected (**outbound requests**)
– those in partner systems (**inbound requests**)

With this information, we can now make a distinction between four basic functions:

- **Outbound send**
- **Outbound receive**
- **Inbound send**
- **Inbound receive**

The possibility of processing transfer data (pre-, post-, and follow-up processing) during a file transfer should be considered an additional function. For FT requests submitted in the local system, no additional protection is necessary since anyone in the local system allowed to initiate FT requests already has access to the available resources. Processing in the remote system does not require any protective measures in the local system either. One function that does require protection in the local system is

- **Inbound processing**

which is initiated from a remote system.

Partner systems also have the option of using the file management functions to view directory or file attributes in their local system, to modify file attributes and to delete files and directories. This results in a further function:

- **Inbound file management**

File management, unlike the other functions, encompasses several different request options, which in turn are partially linked to the functions *inbound send* and *inbound receive*:

Inbound file management function	Prerequisite
Show file attributes	Inbound send permitted
Modify file attributes	Inbound receive **and** inbound file management permitted
Rename files	Inbound receive **and** inbound file management permitted
Delete files	Inbound receive permitted

The protection mechanisms offered by the FTAC function are primarily achieved through the use of admission sets and admission profiles.

2.8.1.2 Admission set

The admission set contains the basic specification of which file transfer functions are permissible. An admission set applies to exactly one login name. When access is attempted under this login name, FTAC checks whether the values set in the admission profile are complied. You can either restrict or extend the specification for the admission set using admission profiles or privileges respectively. If your security requirement is very high, we recommend that you block all inbound functions in your admission set, i.e. all possibilities of reaching your computer from the outside. You can then use the admission profile to permit one or more individual inbound functions for particular partners. You can view admission sets at any time and modify as required to meet your current needs.

Following installation of openFT the entries in the standard FT profile initially apply to all login names. You, as FTAC administrator, must modify this standard FT profile after installation so that it provides the necessary protection for the majority of the login names. If individual login names require greater protection, you can create specially adapted admission sets.

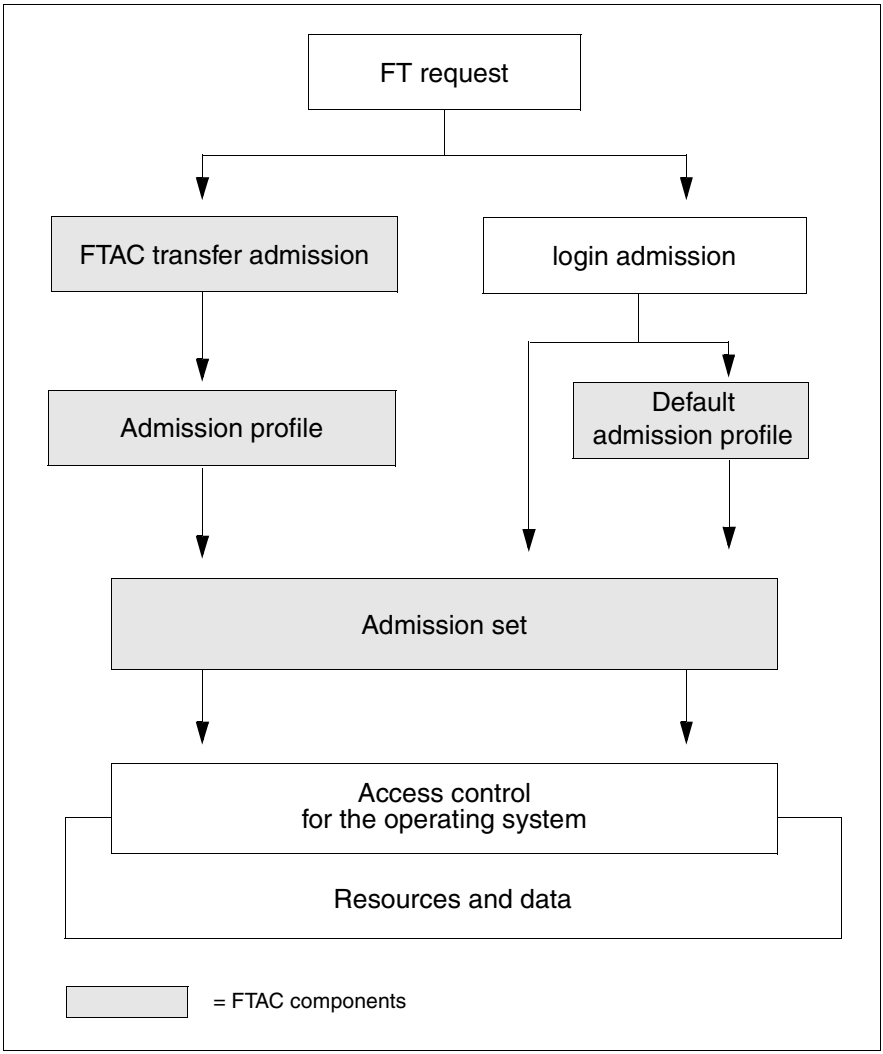
In addition, the FT administrator can assign security levels to the partner systems. When combined with the admission set settings, this makes it possible to prohibit or permit the use of the individual file transfer functions on a partner-specific basis.

2.8.1.3 FT profile (admission profile)

The FT profile (or admission profile) defines the **transfer admission** and the associated **access rights**. The transfer admission is the actual key to your processor. You should therefore treat the transfer admission with the same care as you look after a password. It must be specified in transfer requests instead of a login admission. The standard admission profile for a user ID is an exception. See page 55. Anyone who possesses this transfer admission does have file transfer access to your processor, but, unlike the Login admission, is not free do as he or she please. Which functions you permit are specified with the access rights for this transfer admission. In this way, you can control the conditions under which file are accessed or the follow-up processing commands which are permitted after file transfer. In the most extreme case, you can restrict access to your processor so much only on single profile is available providing access to only one file.

FTAC checks whether the entries in the request conflict with the entries in the FT profile for each file transfer request. If so, the file transfer request is rejected. In this case, only a general error message appears in the remote system. This prevents the definition of the FT profile being established step-by-step on a trial and error basis. A log record which describes the cause of the error precisely is created in the local system.

The following diagram shows the sequences for admission checking with FTAC.



Access check with FTAC

An admission profile includes the following:

– a transfer admission. This transfer admission must be unique. If a request is to work with the FT profile, this transfer admission must be specified. FTAC only permits access rights for this request which are defined in the FT profile. In order to uniquely assign the responsibility for request, it is recommended that a transfer admission be assigned to exactly one person in precisely one partner system.

– if necessary, specification of the partner systems which may access this FT profile.

– Specification of the parameters that may be used in a request. In this way, the access rights are restricted for each person who uses this FT profile.

– If necessary, specification of whether and how long the FT profile is valid.

– A file name prefix. This prefix contains a part of the path name. The user of the profile can only navigate below this specified path name. For example, C:\USR\HUGO\ as a file name prefix on a Windows system means that the user of this profile can only access directories below the path C:\USR\HUGO\. The same principle applies on a Unix system if, for example, /home/user/hugo is specified as a file name prefix. This prevents anyone with this profile to navigate within locked directories or from using the preprocessing function. Note, however, that it is also possible to specify a remote preprocessing command as the file name prefix, in which case, only the parameters for that command would then need to be specified in the request.

You can store various FT profiles.

You are always free to carry out the following operations on FT profiles:

– **Modify**
  and thus adapt the profile to current requirements.

– **Lock**
  In this case, a request with the locked profile is rejected on account of the invalid transfer admission. If you want to use the FT profile again, you must first unlock it.

– **Delete**
  You should limit the number of your FT profiles by deleting profiles which you no longer require.

– **Privilege** (system-dependent)
  In special cases, FT profiles can also utilize a function that has been locked in an admission set. In order to do this, the FT profile must be assigned a privilege by the FTAC administrator.

You may display information about your FT profile at any time.

Standard admission profile

You can set up a standard admission profile for each user ID.

This profile is only intended for certain use scenarios, such as when an FTAM partner has to specify the transfer admission in a fixed structure (user ID and password) for inbound access and you nevertheless wish to specify certain settings, such as a filename prefix.

Unlike a normal profile, a standard admission profile has no FTAC transfer admission, because access is controlled implicitly using the user ID and password. On the other hand, this profile allows most of the normal parameters to be set, such as the permitted FT functions, a filename prefix or the write mode. You cannot set the expiry period, whether or not the profile is locked and whether the profile is private or public.

A standard admission profile must be set up explicitly and a maximum of one standard admission profile can be set up for each user ID.

2.8.1.4 Effects of an admission profile

The following table contains possible restrictions to the access rights in an FT profile in the left-hand column, and the entries for the file transfer request required for the partner system in the right-hand column. Some differences apply to a standard admission profile. See the section “Standard admission profile” on page 55.

Entry in the FT profile	**Entry in the file transfer request**
Transfer admission	The transfer admission addresses the admission profile. If the user ID and password are specified, it is only possible to address the standard admission profile of the user, if this has been defined.

Entry in the FT profile	**Entry in the file transfer request**
Transfer direction restricted	The parameter specified must be the opposite of the entry in the FT profile. If the profile contains transfer direction “from partner”, the remote system may only send data to the local system, and for transfer direction “to partner”, it is only possible to transfer files to the remote system. In contrast, only read access is permitted in the local system.
Partner systems specified	The request can only be issued by the partner systems entered in the profile.
File name specified	The file name must be omitted in the request. If it is a mandatory parameter in the partner system’s file transfer product, it must be assigned the value “*not-specified” (e.g. BS2000/OSD).
Prefix for the file name specified	Only part of the file name is present in the request. FTAC supplements this entry with the prefix defined in the profile to obtain the complete file name. The specification of absolute file names or leaving the directory using “..” is prevented by FTAC.
Processing prohibited	No processing may be requested for your processor.
Processing specified	No processing may be requested for your processor.
Prefix/suffix for follow-up processing specified	Only the part of the follow-up processing which is not defined in the profile may be specified in the request. FTAC supplements this entry to produce the complete follow-up processing command. If no follow-up processing is specified in the request, none is carried out.
Write mode restriction	The request is executed only if it complies with this wirte mode.
Force or forbid encryption	The request will only be carried out if it corresponds to the presets in the admission profile.

Migrating admissions

The FTAC administrator can store both complete admissions as well as individual admission records and profiles in a file (migration). You can then take from the file as required.

2.8.1.5 FTAC administrator

openFT offers the FTAC function for platforms ranging from PC to mainframe. On some stand-alone system the user is responsible for all administrative tasks, whereas large multi-user systems, such as mainframes, offer a multitude of administrative tasks as a centralized service. The FTAC function offers options for these “administration scenarios” by giving, for example, the user of openFT for BS2000/OSD, z/OS, Windows systems or Unix systems the possibility to rely on his or her FTAC administrator. The FTAC administrator, who is not necessarily identical to the FT administrator, also specifies the security framework for his or her system in the form of a standard admission set which is applicable to all users. The individual user then has the option of customizing the security mechanism set by the administrator to meet individual requirements, or to accept the setting made by the FTAC administrator as the lowest security level for his or her system.

2.8.2 Encryption for file transfer requests

When connecting to openFT partners that support the AES algorithm (e.g. openFT V8.0 and higher), then RSA/AES encryption algorithm is used for the request description data and user data instead of the previous RSA/DES algorithm.

To do this, openFT by default uses a 768-bit RSA key and, as of Version 11, a 256-bit AES key. If connections are established to older versions, this 256-bit AES key is either negotiated down to a 128-bit AES key or encryption is negotiated down to DES. Provided that performance considerations permit this, it is also possible to use a 1024-bit or 2048-bit RSA key. The FT administrator must set this in the operating parameters.

For encryption in file transfer requests, a distinction must be made between request description data and user data.

The encryption of the user data is only possible if this function has been enabled with the corresponding module (openFT-CR). This product is subject to export restrictions.

The encryption of user data is only available for data transfer with openFT partners.

Encryption of request description data

Request description data contain security-relevant information, such as addresses and passwords which give access permissions. The encryption of request description data is agreed automatically between the partner systems when a connection is set up, provided both partners support encryption. Otherwise the request description data is transferred unencrypted.

Encryption of user data

Stricter requirements for data security are satisfied by the option of encrypting user data as well. With openFT you can

- purposely request an encrypted transfer of your user data during outbound requests
- force or forbid encryption of user data using an admission profile during inbound requests.

If your FT partner does not offer this capability, or it does not adhere to the presets in the admission profile, then the request will be denied.

Please note that the overhead required for data encryption produces a trade-off with system performance at the partner.

It is possible to control encryption in the admission profile:

- Encryption can be explicitly forced, for example, for requests requiring an especially high degree of security. Requests with unencrypted user data will be denied.
- Encryption can be explicitly forbidden, for example, for requests requiring a lesser degree of security, where performance is key. Requests with encrypted user data will be denied.

The mechanism for active encryption of user data is a separate delivery unit and must be released explicitly due to legal requirements.

2.8.3 Logging openFT operations - the logging function

Prevention of unauthorized access and protection of data inventories is just one security aspect. The complete documentation of the access check and the file transfer requests also puts you in a position to check your security network at any time and detect any leak. The logging function of openFT is the most suitable tool for doing this. It is activated as default and logs all information relating to file transfer requests, irrespective of whether the initiative lies in the local or remote system and whether the transfer was successful or not. The **log records** are written into the corresponding file. The scope of logging can be set as appropriate.

The logging function also serves as a basis for detecting break-in attempts. In addition, it may be used to obtain and evaluate performance data (see also the section “Further processing of openFT data” on page 48).

Log records

If your local system is protected by FTAC, FTAC first checks all accesses to your system and logs the result in an **FTAC log record**. If the access check is negative, FTAC already rejects the request. If the access check is positive, the following applies:

- In the case of a file transfer request (and if the request materializes), an **FT log record** is subsequently written indicating whether the request was executed successfully or why it was cancelled. This means that there can be two log records for one transfer request.
- In the case of a remote administration request, an **ADM log record** is written indicating whether the request was executed successfully or why it was cancelled.

You may display log records relating to your login name at any time, either in abbreviated form or with all data. You may also display only particular log records. e.g. all log records for a certain partner system.

The log record provides the following information:

– Type of log record (FT, FTAC or ADM)
– Date and time when the log record was written
– A reason code which informs about the success or failure of the request
– Name of the partner system
– Direction of file transfer
– Identification of the initiator for outbound
– Name of the file in the local system

Log records of other login names can only be viewed by the administrator.

Logging request with preprocessing / postprocessing

For security reasons, only the first 32 characters (or 42 characters for *ftexecsv* preprocessing) of a preprocessing or postprocessing command are recorded in the log record. The user can influence which command parameters will appear in the log file by arranging the call parameters accordingly or by entering spaces in the list of parameters.

Specifying the scope of logging

the FT administrator has the following selection options for the FT log record:

– never log
– log only errored file transfer requests
– log all file transfer requests

All file transfer requests are logged as default.

As FTAC administrator, you have the following selection options for the FTAC log record:

– log only rejected FTAC access checks
– log only modified file management requests and rejected FTAC access checks
– log all FTAC access checks

All FTAC access checks are logged as default.

The FT administrator can choose between the following options for the ADM log record:

– never write a log record
– only log failed remote administration requests
– only log remote administration requests that modify data
– log all remote administration requests

By default, all remote administration requests are logged.

Saving and deleting log records

Only the FT administrator, the FTAC administrator and the ADM administrator are permitted to delete a log record. Log records should be saved at regular intervals (ideally using a cyclical job). During this, the output of the *ftshwl* command, not the log file itself, should be saved. Afterwards, log records are removed from the current log file.

The benefit of this is, first, that the log records provide a complete record of FT operations which can be maintained for long periods, and second, that the log file does not assume unnecessarily large proportions, which saves CPU time when accessing the records.

2.8.4 Authentication

If data requiring an extremely high degree of security is to be transferred, it is important to subject the respective partner system to a reliable identity check (“authentication”) before the transfer. The two openFT instances engaged in the transfer can perform mutual checks on one another, using cryptographic resources to determine whether they are connected to the “correct” partner instance.

An expanded addressing and authentication concept is therefore supported in openFT (versions 8.1 and later) for Windows and Unix systems, or V9.0 for BS2000 and for z/OS. It is based on the addressing of openFT instances via network-wide, unique IDs and the exchange of partner-specific key information.

Instance identification

Each openFT instance that works using authentication, must be assigned a network-wide, unique instance identification (instance ID). This is a name, up to 64 characters long, which, as a rule, should correspond to the DNS name of the openFT instance. The unique instance ID must not be case-sensitive. The FT

administrator defines these IDs for the local system using an operational parameter. Instance IDs of partner systems are stored in the partner list. openFT administers the resources assigned to these partners, such as request waiting queues and cryptographic keys, with the aid of the instance IDs of the partner systems.

Key administration

The FT administrator can prepare a maximum of three RSA key pair sets for each local openFT instance, from a private and a public key respectively. The public keys are stored under the following name at the following location:

syspkf.r<key reference>.l<key length> in directory *config* of the openFT instance. In the case of the default instance, *config* is in the directory */var/openFT/std*).

The key reference is a numerical designator for the version of the key pair, the key length is currently 768 or 1024 or 2048 bits. The public key files are text files that are created in the character code of the given operating system, i.e. as standard:

– BS2000/OSD: value of the system variable HOSTCODE
– z/OS: IBM1047
– Unix systems: ISO8859-1
– Windows systems: CP1252

In order that one's own openFT instance can be authenticated in the partner system, the appropriate public key must be made available to the partner system. This should take place via a secure path, for example by

– distribution by cryptographically secure e-mail
– distribution on a CD (by courier or registered mail)
– distribution via a central, openFT file server, for which you have a public key.

If the key files between Windows or Unix systems and BS2000 or z/OS are exchanged, you must ensure that these files are re-coded (e.g. by transferring them as text files via openFT).

In the Unix system, the public keys of the partner systems are stored in the directory *syskey* of the instance directory (in the case of the default instance, *syskey* is located under */var/openFT/std*). The file name chosen must be the instance ID of the partner in lowercase characters.

2.9 Using openFT in a cluster

In openFT you can simultaneously execute more than one openFT instance on a single host. This allows you to switch to the openFT functionality on a different computer that is already running openFT when your computer fails.

openFT commands that can be called during preprocessing, postprocessing or follow-up processing execute in the same instance as the request that initiated the preprocessing, postprocessing or follow-up processing.

To set up in which Instance openFT commands should run,there are two ways to specify with which instance openFT is to run:

– Via the openFT Explorer

  If there is more than one instance, then a list appears in the openFT Explorer from which you can select an instance.

  This setting then applies to all commands and menu options which are entered via the openFT Explorer.

– Via the *ftseti* command

  This setting then applies to all commands which are entered via the shell.

Furthermore, you can output information on the instances with the *ftshwi* command.

You will find a detailed description of the commands in the command chapter.

2.10 Switching language interfaces

The language is not queried during installation. Instead, the *LANG* environment variable of the administrator installing is evaluated and set as the default language. This value can be changed as follows:

- The openFT administrator can change the default setting with the *ftlang* tool. Only the setting specified via the *ftlang* tool is relevant for the output of the man pages.
- Each user can change his or her own language setting using the OPENFTLANG environment variable. The user must enter the first two letters of the language setting in the *LANG* variable (*de* or *en*) and then export the environment variable.

  Example

  `OPENFTLANG=de; export OPENFTLANG` corresponds to (for example): `LANG=De_DE.88591,De_DE.646`, etc.

  or

  `OPENFTLANG=en; export OPENFTLANG` corresponds to (for example): `LANG=En_US.ASCII,En_US.88591`, etc.

The following table shows the effects of setting (or not setting) OPENFTLANG and LANG:

OPENFTLANG	**LANG**	**Result**
Not set or empty	Not set or empty	Default setting
Not set or empty	Invalid value	Default setting
Not set or empty	Valid language (German or English)	Language set in LANG
Invalid value or a language that is not installed	Not evaluated	Default setting
Valid value (de or en)	Not evaluated	Language set in OPENFTLANG

3 File transfer and file management

File transfer with openFT is initiated by a file transfer request. In the file transfer request, you make entries to specify the partner system, the transfer direction, the file name and file properties. Given the variety of hardware and software platforms supported, the values specified are subject to various different conventions applicable to the operating systems involved in file transfer. Which files can be transferred between two computers depends on whether the file transfer partners are running identical operating systems (homogeneous link), or different operating systems (heterogeneous link). If a partner using the FTAM functionality is involved in file transfer, the link is a heterogeneous one as a rule. The file management offered by openFT allows you to delete, rename files, or change file attributes before or after file transmission.

The use of the FTAC functionality offers you not only security benefits, but also allows you to make your file transfer operating system independent (see the section “Features of the FTAC function” on page 49), provided the appropriate FTAC settings exist on the processors involved in the file transfer.

Entries for file transfer requests

The following sections give you an overview of the entries you have to make for a file transfer request. They are divided into a local, a remote and an optional part. In the local part, you specify the local file name, if necessary, with the directory name and the file passwords. In the remote part, you define the remote file name, the partner computer and the access to this processor (login name and, if antecessor, the account number and password or transfer admission). In the optional part, you have the option of specifying transfer modalities, such as file types, and follow-up processing requests, for example.

3.1 File names

The description below provides an overview of the system-specific conventions for entering file names, regardless of whether a local or remote file name is involved. By using the FTAC functionality with an appropriate definition in the FT profile, you can avoid having to enter all or part of the file name (see the section “FT profile (admission profile)” on page 52). In other words, the parts of the file name defined in the FT profile need not specified in the file transfer request again.

3.1.1 Unique file names for receive files

The following applies to all file names:

If a file name ends with %unique or %UNIQUE, this string will be replaced by another string, which varies with each new call.

This string is 14 characters long in Unix systems, 18 characters long in Windows systems, 22 characters long in BS2000 systems and 15 or 8 characters long (for libraries) in z/OS systems. If the receiving system is a Unix or Windows system, a suffix may follow %unique or %UNIQUE separated by a dot, e.g. "file1%unique.txt". This suffix must not contain any dot.

Only the converted file name appears in logs and messages.

In follow-up processing, even from FTAC profiles, the variable %FILENAME is replaced by the already converted file name (but without any extension due to a file name prefix that may have been defined in the FTAC profile).

Possible applications include:

- sending a file and then printing and deleting it
- sending a file to an “intermediate system” in order to forward it from there and then delete it on the intermediate system

Note that the specification of %unique is not meaningful for send files or in the case of file extensions.

Remote file names in receive requests that begin with a vertical bar (|) are interpreted as preprocessing commands, provided the remote partner supports the preprocessing function.

3.1.2 BS2000/OSD file names

Format for BS2000 (DMS)	Meaning
:cat:$user.filename	:cat: Optional specification of catalog ID; Available characters restricted to A…Z and 0…9; max. 4 characters; must be enclosed in colons; Preset is the catalog ID assigned to the login name in the entry in the user catalog.
	$user Optional specification of login name; Available characters A…Z, 0…9, $, #, @; max. 8 characters; must not start with a digit; $ and the dot must be entered; Preset is the catalog login name under which the file is accessed.
	filename File name can be split up into several subnames: $name_1$[.$name_2$[...]] $name_i$ contains no blanks and must start or end with a hyphen; File name can be up to 41 characters long, must not start with $ and must contain at least one character in the range A…Z.
:cat:$user.group (gen-no)	:cat: see above $user see above group Name of a file generation group Available characters: A…Z, 0…9, $, #, @. Brackets must be specified. Length up to 41 characters. (gen-no) (*abs) absolute generation number (1..9999); * and brackets must be specified. (+/-rel) relative generation number (0..99); Signs and brackets must be specified.

Format for BS2000 (DMS)	Meaning
:cat:$user.lib/type/element	:cat: see above $user. see above lib Library name; the rules for BS2000 DMS file names apply.
	type Element type; Alphanumeric name, 1 - 8 characters in length.
	element Element name; The rules for LMS element names apply; element can be up to 64 characters in length, must not begin with $, and must include at least one character from A…Z.

In the remote BS2000 operands for the POSIX file names, the POSIX file name must be specified as a C string (graphic string) (i.e. enclosed in quotation marks). This is necessary in order to distinguish between uppercase and lowercase in POSIX file names.

Format for BS2000 (POSIX)	**Meaning**
posix-filename	Character string up to 255 characters long. Comprises either one or two dots, or alphanumeric characters and special characters; special characters must be escaped with \. The character / is not permitted. Must be enclosed in single quotes if alternative data types are allowed, separators are used or if the first character is ? or !. The POSIX file name must be preceded by a POSIX path name.
posix-pathname	Input format: [./]$part_1$/.../$part_n$ where $part_n$ is a posix file name; up to 1023 characters; must be enclosed in single quotes if alternative data types are allowed, separators are used or if the first character is ? or !. The POSIX path name must begin with / or ./ or consist of at least / or ./

3.1.3 File names in Unix systems

Up to 512 characters, where a distinction is made between uppercase and lowercase. It is recommended that the following characters be avoided in file names:
- ? @ # $ ^ & * ( ) ‘ [ ] \ | ; " < >.

3.1.4 Windows file names

File name here refers to the complete pathname.

Up to 256 characters. The following characters must not be used:
| * ? " < >.

No network drives can be specified for remote file names, either when fetching or sending files. Instead, you can specify UNC names.

UNC names

UNC names (**U**niversal **N**aming **C**onvention) are addresses of shared resources in a computer network. They have the following format:

```
\\hostname\sharename\path\file
```

Either the host name or the IP address, for example, can be specified for *hostname*:

```
\\host1\dispatch\catalogs\winterissue.pdf
```

or

```
\\172.30.88.14\dispatch\catalogs\winterissue.pdf
```

3.1.5 z/OS file names

Format for z/OS	Meaning
':S:first-qual.filename' or :S:filename	Specification for PS dataset :S: Optional prefix to identify a PD dataset first-qual "first level qualifier" Specification of login name (max. 7 chars); possible characters: A…Z, 0…9, $, #, @; must not start with a digit or alias name (max. 8 characters) filename partially qualified file name can be split up into several subnames using dots: $name_1$[.$name_2$[...]] $name_i$ is up to 8 characters long; available characters: A…Z, 0…9, $, #, @; must not start with a digit or "-" The partially qualified file name can be up to 36 characters long Fully qualified name The fully qualified file name (first-qual.filename) can be up to 44 characters long.
':S:first-qual. gen-group. GmmmmVnn' or :S:gen-group. GmmmmVnn	Specification for absolute file generation :S: Optional prefix to identify a PD dataset first-qual See "Specification for PS dataset" for syntax gen-group See filename in "Specification for PS dataset" for syntax Exception: partially qualified file name, up to 27 characters; fully qualified file name up to 35 characters Gmmmm.Vnn absolute file generation mmmm absolute generation number (0000 - 9999) nn version number (00 - 99)

Format for z/OS	Meaning
':S:first-qual. gen-group(rel-gen-no)' or :S:gen-group(rel-gen-no)	Specification for relative file generation :S: Optional prefix to identify a PD dataset first-qual See "Specification for PS dataset" for syntax gen-group See gen-group in "Specification for absolute file generation" for syntax rel-gen-no relative generation number 0 = current generation +/-m = 1 - 99 for partially qualified specification (without first-qual and quotation marks) 1 - 255 for fully qualified specification (with first-qual and quotation marks)
':prefix:first-qual. filename (membername)' or filename (membername)	Specification for PO or PDSE member :prefix: Optional prefix to identify the file organization; can assume the following values: :O: for PO :E: for PDSE :L: for PO or PDSE first-qual Syntax see "Specification for PS dataset" filename Partially-qualified file name of PO or PDSE dataset Syntax see filename in "Specification for PS dataset" membername Name of PO or PDSE member max. 8 characters long, available characters: A…Z, 0…9, $, #, @; must not start with a digit

Format for z/OS	**Meaning**
‘:V:first-qual.filename‘ or :V:filename	Specification for VSAM file of type “entry-sequenced” :V: Optional prefix to identify a VSAM file of type “entry-sequenced” first-qual Syntax see “Specification for PS dataset” filename Partially-qualified file name of VSAM file Syntax see filename in "Specification for PS data set"
‘:prefix:first-qual.filename‘ or :prefix:filename	Specification for a complete PO or PDSE dataset :prefix: Optional prefix to identify the file organization; can assume the following values: :O: for PO :E: for PDSE :L: for PO or PDSE first-qual Syntax see “Specification for PS dataset” filename partially-qualified file name of PO or PDSE data set See filename in "Specification for PS data set" for syntax Exception: maximum length of partially-qualified file name is 34 characters, fully-qualified file name is 42 characters. Thus the maximum permitted file name length is, for both partly and fully qualified specifications, 2 characters shorter than for a PS data set. This is because the name of a temporary data set required to transfer a complete PO or PDSE data set is formed by adding ".U"

Access to files of the z/OS Unix System Services (openEdition files) is supported as of openFT V10 for z/OS. The file names comply with the POSIX conventions.

Format with z/OS	Meaning
filename	Components of an openEdition filename. String up to 255 characters in length. This comprises either one or two periods or alphanumeric characters and special characters. The character / is not permitted.
pathname	openEdition file name Input format: [.]/[$part_1$/.../$part_n$] where $part_n$ is a POSIX file name; up to 512 characters. If the name starts with /, it is interpreted as an absolute path name. If the name starts with ./, it is a “relative” path name and is relative to the directory for the user ID, e.g. /u/userid in lowercase characters/.

3.2 File passwords

If a password applies to a file that is accessed with openFT is password-protected, the password must be entered. In Windows and Unix systems, there are no file passwords.

System	File password
BS2000	1 - 4 character C string (graphic string) or 1 - 8 character X string (octet string) or integer string between 2147483648 and 2147483647
z/OS	1 - 8 alphanumeric characters

3.3 File types

Depending on their file type and the operating system from which they originate, files that can be transferred have different properties, which must be considered during the transfer.

3.3.1 BS2000/OSD files

In accordance with the different file structures, a distinction is made between the following BS000 file types:

- Cataloged files
  - DMS files (these include SAM, ISAM, and PAM files, PLAM libraries and cataloged generations of a file generation group)
  - POSIX files
- Elements of a cataloged PLAM library
  - Printable or user-definable elements of type D, J, M, S and possibly X
  - Elements with BS2000-specific binary code of type C, L, R and possibly X

In order to be able to transfer POSIX files using openFT, POSIX must be started. The POSIX file system essentially corresponds to the layout and structure of the Unix file system.
The following overview shows the relationship between file name syntax and file type in BS2000.

File name syntax	**File type**
Starts with $userid or :catid:$userid and does not contain '/'	DMS file, fully qualified
Starts neither with '/' nor with './' nor with $userid nor with :catid:$userid and does not contain '/'	DMS file, path relative to transfer admission
Starts with '/'	POSIX file, fully qualified
Starts with './'	POSIX file, path relative to transfer admission
Starts $userid or :catid:$userid and contains at least one '/'	Name of a PLAM element, fully qualified
Starts neither with '/' nor with './' nor with $userid nor with :catid:$userid but contains at least one '/'	Name of a PLAM element, path relative to transfer admission

BS2000 files may be located either on common disks or on private disks. For processing of files on private disks, the files must be cataloged and the private disks must be properly connected to the system.

3.3.2 z/OS files

openFT for z/OS can transfer the following types of files:

– PS datasets including absolute and relative file generations
– Members of PO and PSDE datasets (with the exception of object modules and programs)
– VSAM files of type “entry-sequenced”
– openEdition files (files belonging to the z/OS Unix Systems Services)
– Migrated files, i.e. files swapped out with HSM. See also the section “Migrated files” on page 84.

The transfer of these files is performed sequentially. The files can be transferred homogeneously between two z/OS systems or heterogeneously with a non-z/OS system or a non-z/OS system. For homogeneous file transfer, all file types can be mapped to one another. Between z/OS and other platforms (heterogeneous link) it is possible to transfer files if the remote system also supports sequential files. WithBS2000/OSD systems, for example, SAM files and PLAM elements of the appropriate type can be exchanged.

The transfer of complete PO and PSDE datasets can only take place between two z/OS systems.

z/OS files may be located either on common disks or on private disks. For processing of files on private disks, the files must be cataloged and private disks must be properly connected to the system. For the processing of files on private media, the precondition is that the files are cataloged and that the private data medium has been properly connected to the system.

The following files cannot be transferred by openFT:

– Files with the attribute “unmovable” (data organization PSU)

3.3.3 Unix and Windows files

Files in Unix systems and Windows systems, like POSIX files in BS2000/OSD, have no structure and no file attributes that provide information on the coding. Although they have no structure either, Windows files can be distinguished on the basis of their file extensions (e.g. “txt” for text and “exe” for executable files).

For transfer with Windows or Unix systems, you can therefore define the following file types:

– text
– unstructured binary data
– binary data structured in records (user format)

Text format

A file that is sent in text format from Windows or Unix systems, must be a pure text file with a record structure defined by linefeed characters in Unix systems or Carriage Return and linefeed in Windows. The end-of-line character is removed from every line.

During transfers from BS2000/OSD or z/OS to Windows or Unix systems, the end-of-line character is inserted into the sentence length already in the remote system. The text and the sentence lengths are preserved. The maximum sentence length during a text file transfer depends on the operating system.

When communicating with partner systems as of openFT V10, it is also possible to transfer Unicode files; see section “Transferring 7-bit, 8-bit and Unicode files” on page 85.

Tabulator and blank line expansion

During transfers of text files, openFT carries out a tabulator and blank line expansion if necessary. This means that blank characters will be transferred instead of a tabulator, and a line with a blank character will be transferred instead of a blank line. During this, the following cases will be different for openFT partners:

Initiator	Direction	Responder	Expansion (yes/no)
Unix system, Windows system	Send	Unix system, Windows system	no, optional yes[1]
Unix system, Windows system	Fetch	Unix system, Windows system	no
Unix system, Windows system	Send	BS2000, z/OS	yes, optional no[1]
Unix system, Windows system	Fetch	BS2000, z/OS	no (not relevant)
BS2000, z/OS	Send	Unix system, Windows system	no (not relevant)
BS2000, z/OS	Fetch	Unix system, Windows system	yes (at the initiator)
BS2000, z/OS	Send and Fetch	BS2000, z/OS	no

[1]The expansion can be explicitly enabled or disabled in Unix systems and Windows system during the request.

During file transfer with FTAM partners, there is no blank line expansion. Tabulators are expanded during transfers using the character set *Graphic String*, but not in the *General String*. For more detailed information on FTAM character sets, see also section “FTAM files” on page 79.

Binary format

When “Binary format” is specified, it is assumed that the file to be transferred contains an unstructured sequence of binary data. In the receiving system, a file with an undefined record length is generated. The binary data remains the same.

User format

When sending a file, it is assumed that length fields divide up the file into records. The first two bytes of each record must indicate its length, including the length of the record length field. When the file is fetched, this length data is generated in accordance with the actual record lengths in the remote system. The contents of the records are treated like binary data, i.e. not converted.

Both the record structure and the binary data remain unchanged when a file is transferred. The record length fields are stored in all Unix and Windows systems starting with the most significant byte. The maximum permitted record length within a file in the user format depends on the operating system.

3.3.4 FTAM files

You can exchange the so-called “document types” FTAM-1 (for text files) and FTAM-3 (for binary files) with FTAM partners.

The file structure and contents of these FTAM files are described in the Kernel group in “contents-type”:

- constraint set
  The constraint set describes the file structure. The subset of the FTAM standard selected by the functional standard ISO/EN 10607-3 permits only the value *unstructured*.
  The *constraint set* also specifies the actions which are permissible with the file on the basis of the structure of the file. For unstructured files, read, overwrite, extend and delete operations are permitted. Together with the *permitted actions*, the *constraint set* restricts the set of possible actions on a file.

  document type
  describes the actual contents of the file. ISO/EN 10607-3 requires support of FTAM-1 (unstructured text) and FTAM-3 (unstructured binary) for files with binary contents. The string format (*string significance*) can be variable (*variable*), fixed (*fix*) or not significant for storage (*not significant*). Furthermore, a maximum length of the string (*maximum string length*) can also be defined.

  In the case of text files (FTAM-1), the *universal class number* specifies the characters present in the text:

– *GraphicString* can contain all graphical character sets (G sets) and escape sequences can be used to switch between character sets (see ISO 2022).

  openFT sets the character set to ISO 646 IRV (or ASCII IRV or ISO 8859-1 G0 set) plus ISO 8859-1 G1 set which broadly covers the characters used in the European languages. When two partners interconnect with openFT as of V10, the character set for file transfer is set to UTF-8.

– *GeneralString* may contain not only graphical characters but also control character sets (C sets) which can also be switched.

– *VisibleString* contains only graphical characters from ISO 646 IRV.

– *IA5String* contains graphical characters from ISO 646 IRV and control characters from ISO 646 (C0 set).

3.3.5 Transfer of various file types

Besides complete transfer of the contents of a file, file transfer also aims at producing an authentic representation of the file structure. If identical structures are mapped to each other, as is the case with homogeneous links, authenticity is achieved without any problem, i.e. the binary code and the character representation are identical in the send and receive system. With heterogeneous links, however, it is usually not possible to obtain the binary code and the character representation in the receive system unchanged. For this reason, a distinction is made between text and binary transfer for file transfer with openFT. More details on file transfer with FTAM partners ca be found in the section “Special points for file transfer with FTAM partners” on page 108.

Text transfer

Text transfer is character-oriented, i.e. the presentation of the characters is retained. This applies both to characters in single-byte code such as ISO 8859 and to Unicode characters which are represented by multiple bytes. The record structure of the text file is matched to the system conventions of the receive system when the file reaches the receive system.

The “useful data” of a file to be sent per text transfer must not contain any characters which the receive system could interpret as control characters, e.g. X‘15‘ (EBCDIC linefeed) and X‘0A‘ (ASCII linefeed).

In the table below, the local system is always a Unix system.

Record structure in receive system	**Local system**	**Remote file system**	**Direction ← / →[1]**	**File type**
system-conformant (in the usual manner in the receive system)	Unix based	all systems	← / →	Standard text
	Unix based	Unix system, Windows	← / →	Standard text binary

[1] ← = fetching, → = sending

Binary transfer

Binary transfer is carried out such that the coding (binary representation) of the characters is retained. The design of the record structure can be controlled. In this way, openFT matches the record structure with the record structure of the receive system (system-conformant record structure). With the original record structure, the structure of the send system is retained. Furthermore, it is possible to employ your own system-dependent record structures using the FT-specific user format.

It is not possible to fetch binary format files with fixed length or variable length records using the FTP protocol. In particular, this also applies to the output of file transfers with preprocessing on BS2000 or z/OS and the output from commands executed using *ftexec* on BS2000 or z/OS. In this case, you must either transfer files in text format or use a different transfer protocol (openFT).

In the table below, the local system is always a Unix system.

Record structure in receive system	Local system	Remote file system	Direction ← / →[1]	File type
system-conformant (in the usual manner in the receive system)	Unix based	Unix system, Windows	← / →	Standard text binary
original record structure (in the usual manner in the send system)	Unix based	DMS, PLAM, z/OS	→	binary
	Unix based	POSIX, Windows, VMS	← / →	binary
User format (system-independent)	Unix based	DMS, PLAM, POSIX, z/OS	←	user
No record structure (i.e. the record structure is possibly lost)	Unix based	DMS, PLAM, z/OS	←	binary

[1] ← = Fetching, → = Sending

ISAM and PAM files can be transferred between BS2000 systems and other systems as follows:

– in transparent format, see page 83

– by specifying the target format, see the section “Heterogeneous transfer of PAM and ISAM files” on page 83

Record by record transfer

When transferring DMS files between Unix or Windows and BS2000 systems the structure of records in files can be important. If files are transferred from a Unix or Windows system to a DMS file, then you must increase the maximum record length with the *-r* option if the default value of 2000 bytes is not enough.

Transfer with transparent file format

A special case is the transparent file format. This file format provides you with the option of passing through any BS2000 files over a variety of FT platforms to a BS2000 system, while retaining their original file attributes. This procedure is useful for distributing BS2000 files from a Unix based server or Windows server to BS2000 systems, for example. From the point of view of the intermediate processor, the files received, which cannot be used by this processor, are binary files. These files are then set up on the receive processor with their original attributes by openFT for BS2000/OSD.

Heterogeneous transfer of PAM and ISAM files

You can transfer BS2000 PAM files onto a foreign system such as a Unix or Windows system or to z/OS and then retrieve them to BS2000 and store them there as PAM files. The foreign system can also have the initiative for this request. You can also transfer ISAM files from a BS2000 systems onto a foreign system. In all cases, the prerequisite for this is that openFT as of V11 is running on the foreign system.

To do this, proceed as follows:

- Transferring a PAM file from BS2000 to a foreign system
  Specify "sequential" as the target format in the transfer request.
- Storing a binary file from a foreign system as a PAM file in BS2000
  Specify binary as the file format and "block-structured" as the target format in the transfer request.
- Transferring an ISAM file to the foreign system
  Specify "sequential" as the target format in the transfer request. The ISAM keys are integral parts of the records that are read and are therefore transferred with the file. However, they no longer have any function as index keys. The record format of the target file is to be the same as that of the ISAM file. The format used is compatible with FTP-BS2000.

3.3.6 Migrated files

openFT can access migrated files in BS2000/OSD and z/OS. This means that you can view the properties of such files, and transfer, delete or overwrite them. To do this, openFT as of V10 must be used in the system involved. The following applies to the mainframe systems used:

- In BS2000 systems, the file must be a DMS file. It is not possible to directly transfer individual elements of a migrated library. To do this, the migrated library must first be read in. This can, for instance, be done during preprocessing and postprocessing or using /EXEC-REM-CMD or *ftexec*.
- In z/OS systems, z/OS as of V1.7 must be used, because the necessary values are only returned at the system interface as of this version.

3.4 Transferring 7-bit, 8-bit and Unicode files

In computers with different operating systems, the individual characters, letters and digits are represented internally ("coded") in different ways. In addition, it is possible to use different character sets in these various systems. The content of a text file is interpreted differently depending on the character set used and is output accordingly on the screen or at the printer.

openFT makes it possible to assign various single-byte character sets (7-bit and 8-bit) as well as multi-byte character sets (Unicode) to text files.

3.4.1 Code tables and coded character sets (CCS)

The concept of so-called "Coded Character Sets" (CCS) is supported for openFT partners. A CCS defines a character set and the coding of these characters in the file. A CCS is assigned a name of up to 8 characters in length via which the CCS can be addressed.

In Unix and Windows systems and in z/OS systems, the standard character set is defined via openFT operating parameters. In BS2000/OSD systems, the character set defined in the system settings is used by default (BS2000/OSD: HOSTCODE system variable). However, in BS2000/OSD, it is also possible to assign a file a specific CCS via the catalog entry, see also "openFT for BS2000/OSD - User Guide".

Moreover, for each individual file transfer, you can specify a CCS separately for the local and remote files, see section “Specifying the CCS on a transfer request” on page 86.

Frequently used example CCS's are:

ISO88591
: Character set in accordance with the definition contained in ISO standard 8859-1, ASCII-oriented coding in accordance with ISO standard 8859-1.

EDF041
: Character set in accordance with the definition contained in ISO standard 8859-1, EBCDIC-oriented coding in accordance with Fujitsu definition DF04-1.

IBM1047
: Character set as defined in ISO 8859-1. IBM1047 is an EBCDIC-based encoding compliant with the IBM definition IBM1047 and used as default in z/OS systems.

UTF8 The character set is Unicode, the UTF-8 multi-byte coding defined in the Unicode standard is used.

UTF16 The character set is Unicode, the UTF16 16-bit coding defined in the Unicode standard is used.

CP1252
The character set is a Microsoft-defined superset of the character set specified in ISO standard 8859-1. The codings of CP1252 and ISO 8859-1 are identical for the shared characters from the ASCII 7-bit character set. The other characters defined by Microsoft (including the Euro symbol) are present in the code range 0x80-0x9F which is not used by ISO 8859-1.

3.4.2 Specifying the CCS on a transfer request

When transferring text files, you can specify a request-specific CCS for both the local system and the remote system:

– *ft -lc* / *ncopy -lc*
Specifies the CCS for reading or writing the local file.

– *ft -rc* / *ncopy -rc*
Specifies the CCS for reading or writing the remote file.

The local/remote CCS can also be specified via the openFT Explorer.

If the remote file is a BS2000 file to which a CCS name has already been assigned via the catalog entry then you may not specify a CCS name that is different from this.

The remote CCS name is only supported for the openFT protocol and for partners as of V10.

If the local or remote CCS name is omitted then the default settings for the relevant system apply:

– openFT operating parameters in a Unix system, Windows system or z/OS system,

– in a BS2000 system, the CCS corresponding to the file's catalog entry (if present), otherwise the HOSTCODE system parameter.

In z/OS, a particular CCS can be assigned to files on the basis of a setting in the FT parameter library.

Caution!
If you save the file in a character set which is not a superset of the character set originally used for the file then information is lost! All characters that cannot be mapped to the newly assigned character set are represented by a replacement character. This type of conversion cannot be undone without data loss!

3.4.3 Data conversion

The type of data conversion depends on the openFT version that is used on the partner system.

Data conversion in the case of partners as of V10

Depending on the code class (ISO 8859 or DF04) and code variant n (n=1...10, 13, 15) of the local CCS, openFT as of V10 sends the data encoded in ISO 8859-n, DF04-n or UTF-8.

This has the following effect depending on the partner system:

– Files in Unix and Windows systems to which an ISO8859n CCS is assigned are no longer recoded in the event of send requests to Unix or Windows systems. In the case of transfers between Unix or Windows systems no recoding is now performed for the transfer itself if the same ISO8859n CCS has also been assigned for the target file.

– In the case of transferring files belonging to the code classes ISO 8859 or DF04 between Unix and Windows systems and BS2000 or z/OS, recoding is performed at the receiving system (if necessary).

– UTF-8 files are recoded at the receiving system (if necessary). Files to which a CCS is assigned that belongs neither to the ISO 8859 code class not to DF04 are recoded into UTF-8 at the sending system and into the CCS of the target file at the receiving system (if necessary).

– UTF-16 files are recoded into UTF-8 at the sending system and into UTF-16 at the receiving system (if this is requested).

– UTF-16 files generated by openFT possess the endian model and line break convention (LF or CRLF) appropriate to the platform in question.

– UTF-8 files generated by openFT possess the line break convention (LF or CRLF) appropriate to the platform in question.

Data conversion in the case of partners < V10

The transferred data is coded in DF04-n. I.e. when file transfer is performed with openFT partners, the data is transferred in EBCDIC format (corresponds to CCS DF04-n). EBCDIC is used, for example, in BS2000/OSD. For this reason, openFT always converts text files when transferring to and from openFT partners:

– when retrieving a file from EBCDIC to ISO 8859,

– when sending a file from ISO 8859 to EBCDIC.

Special characters or alternate representations not defined in ISO 8859 are not converted during code conversion. Files containing such characters should be transferred as binary files, and converted using a user-defined code conversion routine.

In the case of data transfer handled using the FTAM functionality, it is assumed that ISO 8859 is used for the transfer and for the local file with connections between third-party products and openFT partners < V10. No local recoding is therefore performed.

Text format

When sending, openFT assumes that the file to be sent is a pure ISO 8859 text file, which is structured as records separated by carriage returns/line feeds.

In certain situations, a conversion takes place, i.e. tab characters are expanded into blanks and end-of-line characters are eliminated. Depending on the situation (inbound, outbound) and the participating partners, the following applies:

– Inbound requests:

  Conversion to Unix or Windows is not available for send or receive operations on the inbound side.

– Outbound requests issued by a Unix or Windows system:

  Conversion never occurs when receiving requests.

  Request-specific conversion (*ft -tb=* and *ncopy -tb=*, TabExpansion) is possible on send operations. By default, send operations to BS2000, OS/390 or z/OS partners are converted. In all other cases conversion does not take place.

– Outbound requests which are issued in a BS2000, OS/390 or z/OS system:

  Conversion never occurs when sending requests.

  Conversion occurs when receiving requests, depending on the partner, i.e. conversion occurs for a Unix or Windows partner but not for BS2000, OS/390 or z/OS partners.

Binary format

openFT assumes that the file to be transferred contains an unstructured sequence of binary data. In the receiving system, a file is created with an undefined record length. The binary data is retained.

User format

When sending, openFT assumes that the file to be sent is structured by length fields in records. The first two bytes of each record must contain the length of that record, including the length of the record length field. When retrieving, openFT generates these length specifications in accordance with the record lengths in the remote system. The record contents are handled as binary data, i.e. not subjected to code conversion.

The record structure and the binary data are retained during transfer. The highest-order byte of the record length field is stored first in a Windows system.

There is no point using user format for FTP partners since the record structure is lost.

A different mechanism is used between FTAM partners (see section “Virtual filestore” on page 108).

3.5 Entries for the remote system

With the entries for the remote system, you define the partner system and inform it of your transfer admission for a login name in the partner system. The FTAC functionality allows you to restrict the partner system, the transfer direction and the access to a login name to meet your specific security needs.

3.5.1 Defining the partner computer

The partner system is the remote system with which files are to be exchanged. By specifying the transfer direction or the syntax in the *ft*/*ncopy* command you stipulate whether the partner is to send or to receive files. You address the partner system via a partner name or its partner address ("**dynamic partners**").

The FT administrator may deactivate the use of partner addresses for security reasons. In this case, you may only use partner names from the partner list.

Partner name

A partner name is a name of 8 characters or less which is assigned by the FT administrator when including a partner system in the partner list. This approach should primarily be used for partner systems which are frequently communicated with.

Partner address

If the FT administrator has not assigned a partner name or if you do not know the name, you can address a partner host using the partner address. A partner address has the following structure:

[protocol://]host[:[port].[tsel].[ssel].[psel]]

host (= computer name, see page 91) is mandatory; all other specifications are optional. In many cases, the other specifications are covered by the default values, so that the host name suffices as the partner address, see “Examples” on page 94. Final ’.’ or ’:’ can be omitted.

The individual components of the address have the following meanings:

protocol://

Protocol stack via which the partner is addressed. Possible values for *protocol* (uppercase and lowercase are not distinguished):

openft openFT partner, i.e. communication takes place over the openFT protocol.

ftam FTAM partner, i.e. communication takes place over the FTAM protocol.

ftp FTP partner, i.e. communication takes place over the FTP protocol.

ftadm ADM partner, i.e. communication takes place over the FTADM protocol for remote administration and ADM traps.

Default value: **openft**

Exception: if a global name from the TNS is used for *host* and a presentation selector is assigned to this name in the TNS then **ftam** is the default value.

host

Computer name via which the partner is addressed. Possible entries:

- internet host name (e.g. DNS name), length 1 to 80 characters
- Global name from the Transport Name Service (TNS), up to 78 characters long, with full support for the 5 name parts. In this event, the following applies:
  - TNS must be activated (*ftmodo -tns=y*) to allow a global name from the TNS to be used in requests. In this case, the TNS name takes precedence over the Internet host name.
  - The partner address must end with *host* and must not contain any other address components, such as *port*, *tsel* etc.
  - *ftp* is not permitted for *protocol*, as openFT-FTP does not support TNS operation.
  - If the TNS entry contains a presentation selector for this global name, only *ftam* is permitted for *protocol*.
  - If the TNS entry does not contain a presentation selector, *ftam* is not permitted for *protocol*.

– IPv4 address with the prefix %ip, i.e. for example %ip139.22.33.44
You should always specify the IP address with the prefix %ip since the host name is immediately treated as the IP address. Omitting this prefix results in performance impairments since in this case a search is initially performed in the TNS and then in the file /etc/hosts.
The IP address must always be specified as a sequence of decimal numbers separated by dots and without leading zeros.

– IPv6 address with the prefix %ip6, i.e. for example
`%ip6[FEDC:BA98:7654:3210:FEDC:BA98:7654:3210]` (ipv6) or
`%ip6[FE80::20C:29ff:fe22:b670%5]` (ipv6 with scope ID)

  The square brackets [..] must be specified.

  The scope ID designates the local network card via which the remote partner can be accessed in the same LAN segment. It must be appended to the address with a % character. In Windows systems, this is a numerical value (e.g. 5). On other systems, it may also be a symbolic name (e.g. *eth0*). The scope ID can be identified using the *ipconfig* command.

port

When a connection is established over TCP/IP, you can specify the port name under which the file transfer application can be accessed in the partner system.
Permitted values: 1 to 65535;

Default value: **1100** for openFT partners
A different default value can also be set in the operating parameters using *ftmodo -ftstd=*.

4800 for FTAM partners

21 for FTP partners

11000 for ADM partners

tsel

Transport selector under which the file transfer application is available in the partner system. The transport selector is only relevant for openFT and FTAM partners.
You can specify the selector in printable or hexadecimal format (0xnnnn...).

The specification will depend on the type of partner:

– openFT partner:
Length, 1 through 8 characters; alphanumeric characters and the special characters # @ $ are permitted. A printable selector will be coded in EBCDIC in the protocol and may be padded with spaces internally to the length of eight characters.

Default value: **$FJAM**

– FTAM partner:
Length 1 to 10 characters; a printable selector will be coded as variable length ASCII in the protocol. Exception: T-selectors that start with $FTAM (default value) are coded in EBCDIC and padded with spaces to the length of 8 characters.

All alphanumeric characters and the special characters @ $ # _ - + = and * can be used with ASCII selectors.

Default value: **$FTAM**

Note:
As a rule, **SNI-FTAM** must be specified for Windows partners with openFT-FTAM up to V10. As of openFT-FTAM V11 for Windows, the default value has been changed to **$FTAM** and can therefore be omitted.

ssel

Session selector under which the file transfer application is accessible in the partner system. You can specify the selector in printable or hexadecimal format (0xnnnn...).

Length, 1 through 10 characters; alphanumeric characters and the special characters @ $ # _ - + = * are permitted. A printable selector will be coded as variable length ASCII in the protocol.

Default value: empty

psel

Only relevant for FTAM partners.
Presentation selector under which the file transfer application is accessible in the partner system. You can specify the selector in printable or hexadecimal format (0xnnnn...).

Length, 1 through 10 characters; alphanumeric characters and the special characters @ $ # _ - + = * are permitted. A printable selector will be interpreted as variable length ASCII in the protocol.

Default value: empty

Examples

The partner computer with the host name FILESERV is to be addressed over different protocols/connection types:

Connection type/protocol	Address specification
openFT partner	FILESERV
FTAM partner (BS2000, Windows or Unix system with default setting as of V11.0)	ftam://FILESERV
FTAM partner (Windows system with default setting up to V10.0)	ftam://FILESERV:.SNI-FTAM
Third-party FTAM partner	ftam://FILESERV:102.tsel.ssel.psel
FTP partner	ftp://FILESERV

3.5.2 Transfer admission

The transfer admission consists of the login name, the account number and the password (access via login/LOGON admission). These values are system-dependent. You can, however, also define an FTAC transfer admission that is independent of the operating system and provides a higher degree of access protection.

System	FTAC transfer admission	Login name	Account number	Password
Unix based	8 - 32 characters long C string or 15 - 64 characters long X string	1 - 32 characters	Unix systems do not recognize any account numbers locally	Alphanumeric characters (the length is system dependent), a distinction is made between uppercase and lowercase
BS2000	8 - 32 character long C string or 15 - 64 character long X string	1 - 8 alphanumeric characters	1 - 8 alphanumeric characters	1 - 32 character long C string or 1 - 16 character long X string

System	FTAC transfer admission	Login name	Account number	Password
Windows	8 - 36 characters	1 - 36 characters, possibly with leading domain name (DOM\)	Windows does not recognize any account numbers locally	8 - 32 character long C string or 15 - 64 character long X string
z/OS	8 - 32 character long C string or 15 - 64 character long X string	1 - 8 alphanumeric characters	max. 40 characters, uppercase, digits and special characters $, @, #	1 - 8 alphanumeric characters

Examples

If you do not possess FTAC transfer admission then you can specify the transfer admission for the individual platforms using the following syntax:

- BS2000/OSD:

  ```
  userid,account-number[,'password']
  ```

- Unix systems

  ```
  userid[,,password]
  ```

- Windows systems:

  ```
  userid[,,password]
  ```

  The user ID consists of a user name or, if a user ID in a LAN Manager or Windows domain is accessed, it consists of the domain name followed by an backslash (\) and the user name.

  Remember to escape the backslash on Unix systems (\\).

- OS/390 and z/OS:

  ```
  userid,account-number[,password]
  ```

  The accounting number is optional with more recent z/OS versions.

- FTAM partner systems on which no file transfer product of the openFT product family is used:

  ```
  user-identity,[storage account],filestore-password
  ```

- In the case of other partner systems, your specifications depend on the conventions used in the partner system.

Inbound access using the default FTP client

If you wish to access an openFT server from a standard FTP client, you should note the following:

- Establishing a connection
  If the default listener port 21 is set on the openFT FTP server, enter the following from the shell (Unix systems), from the command prompt (Windows) or on command level (BS2000 and z/OS):

  ```
  ftp hostname
  ```

  hostname is the host name of the openFT FTP server.

  If a listener port other than 21 is set on the openFT FTP server, you need two commands to establish a connection:

  ```
  ftp
  ftp> open hostname port-number
  ```

- Login
  If you log in without an FTAC transfer admission, enter the login data interactively as usual (user ID and any password that is required and/or account number). If you log in using an FTAC transfer admission, enter the FTAC transfer admission under *User* and leave the *Password* empty.

  Example

  ```
  User: ftpuser1
  Password: (empty)
  ```

  With openFT FTP servers as of V11, you can enter the value *$ftac* under *User* and the FTAC transfer admission under *Password*.

  Example

  ```
  User: $ftac
  Password: ftpuser1
  ```

3.6 Options for file transfer

openFT offers the possibility to make additional optional setting for file transfer. You can define individual record lengths, agree syntax rules and file compression, and specify conditions for result messages and access modalities for FTAM partners.

3.6.1 Maximum record lengths

The maximum record length is understood to be the length of the longest record (net record length) not including the record length fields.

In Unix and Windows systems, you can set the maximum length of your file which you wish to transfer as text or record-structured binary file (user format) individually. The prescribed maximum record length must be at least as large as the largest one actually available, otherwise the FT request cannot be executed.

3.6.2 Syntax rules

With the option “Syntax rules”, you can define the procedure to be adopted for the destination file during file transfer. This option can also be defined via FTAC. There are two options:

– to overwrite files, i.e. files are overwritten, provided that the file attribute permit this action, or file that do not exist are created,

– to extend files, i.e. existing files are extended at the end of the file, provided that the file attribute permit this action, or file that do not exist are created,

– to not overwrite files; in this case, existing files are under no circumstances overwritten; rather, the FT request is aborted and an appropriate message output. If the specified destination file does not exist, a new file is created.

Access protection for send and receive files

Please note that the destination file is generally not protected from being overwritten by other users while the time the request is being processed. If the transfer is interrupted, for example, then other users may be able to write to the destination file. Access protection differs in the individual systems:

- openFT for BS2000 uses a file lock which protects the files if the transmission is interrupted and between the time of accepting and processing the FT request. This protection does not apply to library members and POSIX files.
- openFT for z/OS protects send and receive files against simultaneous (write) accesses only if data is in fact being transferred, i.e. if the request is in the ACTIVE state. It follows, that the send and receive files are not protected, if the file transfer has not yet begun or has just been interrupted.
- In other systems, for example Unix and Windows systems, or even BS2000, the user is solely responsible for guaranteeing exclusive access to the files to be transferred in the case of POSIX file or library elements. In theses systems, the file cannot be exclusive openFT, not even during file transfer.

  The user him/herself must therefore ensure that (the data and file attributes) in the file to be transferred are consistent throughout the entire duration of the FT request. This applies to both the send and receive files. The danger of eventual inconsistencies resulting from multiple accesses can be reduced, for example, by means of access restrictions (Unix system: *chmod* command). It is also possible to transfer the file to a different name or to a temporary directory and to rename it or move it to a different directory only after file transfer has been completed successfully using follow-up processing.

3.6.3 Compressed file transfer

Files can be sent using data compression. This shortens transmission times and saves costs. However, do note that compression and decompression produce extra CPU load in the receive processor.

openFT is able to use two compression methods - zip compression (with openFT partners as of V10) and byte compression. Both of these can be used to reduce the volume of data for transfer. However, compressing and decompressing the data increases CPU demand and consequently also the time required for a request before and after data transfer itself.

On "fast" lines (as of approximately 10 Mbit), the overall execution time of a request normally is not significantly improved by compression. On "slow" lines (less than1Mbit), zip compression may help enhance performance. Byte compression is worthwhile when transferring files which contain a large number of byte repetitions (e.g. lists with blanks for column alignment, dumps with

numerous zeros). If the partner does not support compression, openFT transfers the file uncompressed. openFT-FTP supports byte compression as described in RFC959.

Data compression is not supported on links to FTAM partners.

3.6.4 Encrypted file transfer

openFT can send data with encryption if requested by the user (see also the section “Encryption for file transfer requests” on page 57).

openFT generally uses the RSA/AES encryption procedure for request description and user data. In the case of connections to partners with older openFT versions (lower than V8.0) then the RSA/DES procedure is used for encryption.

For legal reasons, the encryption option is not available in all countries, i.e. the encrypted file transfer with foreign partners is not guaranteed in all cases. Data encrypted with openFT cannot be exchanged with FTP and FTAM partners.

Data encrypted by openFT van only be exchanged via the FTP protocol in an outbound direction and only with standard secure FTP partners. No data encrypted by openFT can be exchanged with FTAM partners.

Encrypted file transfer always requires openFT-CR to be installed on the openFT side, i.e. also on the partner system if openFT is running there.

3.6.5 Notifying results

The initiator of a file transfer request can arrange to be notified of the result. The logging function, which is available in a standard form on all platforms, is particularly suitable for this.

Other ways of notifying results are platform-dependent:

– In z/OS and BS2000 systems, a file is created on request by the initiator and can be printed out automatically on success or failure of the file transfer.

– In Unix systems, the result message can be stored in the mailbox of the initiator depending on the result.

3.6.6 Access mode

It is possible to define FTAM-specific file attributes for file transfer with FTAM. The FTAM file attributes that describe the file type must be identical to those specified in the file transfer request. The corresponding attributes are presented in the section “Mapping FTAM attributes to the real file system” on page 114ff.

3.6.7 Preprocessing and postprocessing

The “preprocessing” and “postprocessing” functions make it possible to execute any commands (operating system commands, procedures, etc.) with the aid of a file transfer request in the local and remote systems. The commands are passed to the corresponding system instead of the file name. To do this, the file name must be enclosed in double quotes. The first character is a pipe symbol ’|‘. Then follow the commands, separated by ';' (or '&' or '&&' in Windows systems, in which case the command string must start with *cmd /c*). The maximum length of the pre- and postprocessing command is limited by the maximum length of the file name.

If the characters ’|&’ are specified instead of the pipe symbol, the transfer request is restartable, see page 101.

Preprocessing passes the result to the system’s standard output and postprocessing reads the data from the relevant system’s standard input. However, the standard output/input does not usually support all the file formats possible at the system in question. You can avoid this restriction by using the %TEMPFILE variable instead of the standard output/input. This has the advantage of permitting the use of any required file format.

Pre- and postprocessing are part of the request brackets. The issuer of the request always receives a feedback report on the successful or unsuccessful completion of the pre/postprocessing.

If preprocessing or postprocessing runs in a Unix or Windows system then the following applies:

- During preprocessing the data is by default output to *stdout*. You can, however, also output the data created by preprocessing in a temporary file created by openFT. You can find out the name of this file and pass it to preprocessing with the variable %TEMPFILE. The temporary file is then transferred to the partner system.

- During postprocessing, the data is read from *stdin* by default. In this case, it must possess a format which can be processed by *stdin*. However, it is also possible to address the transferred data explicitly via %TEMPFILE.

You should note the following when using the pre/postprocessing function:

- Preprocessing/postprocessing runs as part of the file transfer operation and under the same transfer admission. These specifications are either explicitly stated in the file transfer request or in a transmission profile's USER-ADMISSION. In the case of follow-up processing, different rights may apply depending on the platform (PROCESSING-ADMISSION).
- If the request is handled via an FTAC profile, the FILE-PROCESSING function must be permitted in the profile or, alternatively, a file name prefix starting with the pipe symbol '|' must be defined.
- When non-restartable pre/postprocessing is involved, the connection to the partner must remain intact until the entire processing session is completed. The connection to a partner system using openFT version V8.0 (or earlier) will be cancelled if generation of the transfer data during a preprocessing session (or processing of the transfer data during a postprocessing session) is not completed within 10 to 15 minutes (depending on the partner system).

Restart capability during preprocessing and postprocessing

During restartable pre- and postprocessing, the data to be transferred between openFT and the processing command is always saved to a temporary file. By this means, the request is divided into 3 phases: preprocessing, transfer, and postprocessing.

The restart capability of a pre- and postprocessing session is brought about when you specify an additional “&” before pre- and postprocessing in the transfer command. During this, requests made with openFT partners behave as follows:

– Loss of connection during preprocessing:
  If the connection is lost during the execution of the preprocessing command, the command is still executed until completion after the connection is lost. If the system is restarted after the command has completed execution, then the temporary file is transferred.

– Loss of connection during transmission:
  In this case openFT performs a restart for the temporary file as is usually the case.

– Loss of connection during postprocessing:
  If the connection is lost during the execution of the postprocessing command, the command is still executed until completion after the connection is lost. If the system is restarted, then all other actions left over that belong to the openFT request are performed (e.g. any follow-up processing or the status report to the partner).

The temporary file is stored in the directory *.openFTTmp* and is deleted only after the command has finished execution (regardless of whether or not the command was successful or unsuccessful).

.openFTTmp is created by openFT if it does not yet exist. It is located in the home directory of the corresponding user. On the local host this user is the user under whose user ID the request was started. On the remote host this user is the user whose user ID was specified or who is the owner of the specified transfer admission.

If there are still restartable requests active when an openFT shutdown is initiated and they are still in the command execution phase, then the shutdown is delayed for up to 10 minutes so that the commands have enough time to execute to completion. During this period, a command to shut down the openFT server remains "pending" and the prompt is not displayed until the server process has terminated.

Server function for remote command execution (ftexec)

One special form of preprocessing is the server function for the remote command execution (*ftexec* command). This command makes it possible to execute commands on a remote system. The exit code and/or the output from *stdout* and *stderr* (Unix or Windows systems), SYSLST and SYSOUT (BS2000) or STDOUT=SYSPRINT und STDERR=SYSTSPR (z/OS) are output at the local computer. *ftexec* thus mimics the execution of the command on the local computer.

3.6.8 Follow-up processing

openFT offers four types of follow-up processing requests:

– Follow-up processing in the local system after successful file transfer

– Follow-up processing in the remote system after successful file transfer

– Follow-up processing in the local system after unsuccessful file transfer

– Follow-up processing in the remote system after unsuccessful file transfer

The conventions of the system on which the follow-up processing is to be performed are decisive for the syntax and processing of the statements and commands. A command sequence can only be processed in the remote system if an FT that supports this function in used in the remote system.

You may specify variables within the command or command sequence for follow-up processing. These are substituted at the start of follow-up processing in the particular system using the values obtained from the file transfer requests. The following table shows which variables can be used for which system.

Variable	Meaning	BS2000	Unix system	Windows	z/OS
%PARTNER	Partner name (long form)	X	X	X	X
%PARTNERAT	Partner name (short form)	X	X	X	X
%FILENAME	File name	X	X	X	X
%ELEMNAME	Element name	X			
%ELEMVERS	Element version	X			
%ELEMTYP	Element type	X			
%RESULT	Request result	X	X	X	X
%JOBCLASS	Job class	X			

In the case of %PARTNER and %PARTNERAT, the partner name found in the partner list is used if it is present in the partner list. If it is not entered in the partner list (dynamic partner) then the partner address is used. In this case, %PARTNER and %PARTNERAT have different effects:

– In the case of %PARTNER, all the address components are used, i.e. including protocol prefix, port number and selectors if appropriate.

– In the case of %PARTNERAT, only the *host* address component is used, see page 91. In addition, all characters apart from letters, digits or periods are replaced by '@'.

You may specify data for follow-up processing both for the local and for the remote system, depending on the version of openFT-Version used. In each case, no more than 1000 characters may be used. The number of characters evaluated depends on the operating system and is stated in the relevant FT description. Please observe that the limit length applies after any necessary translation of variables. The limit of up to 1000 characters can be bypassed by calling a procedure, a shell script or a program from within the follow-up processing. A procedure may contain the command sequence which is to be executed on success or failure of file transfer.

Restrictions apply to links with FTP or FTAM partners, since the FTP or FTAM protocol does not permit transfer of follow-up processing data. Follow-up processing in the FTP or FTAM partner system is possible only if it is stipulated there in an FTAC admission profile. It is always possible to initiate follow-up processing in the local system.

The special form of follow-up processing, *DELETE (not for FTAM partners), is available for requests which move files between hosts. This character string can be specified as follows provided that the remote partner system is running openFT V8.0 or higher:

– as remote follow-up processing for synchronous and asynchronous receive requests,

– as local follow-up processing for asynchronous send requests or with FTP partners.

*DELETE causes openFT itself to delete the sent file in the sending system after the termination of the FT request without it being necessary to start a batch job. However, as in the case of genuine follow-up processing, *DELETE does not form part of the job scope. This means there is no response message indicating whether or not the file has been successfully deleted. “Genuine” follow-up processing can be additionally specified via an FTAC profile.

To avoid undefined file fragments in the event of unsuccessful file transfer, it is useful to delete the receive file via follow-up processing in such cases.

3.7 File management

File management in openFT is possible both in the remote and in the local system.

3.7.1 File management in the remote system

openFT offers the option of managing remote system files from the local system (file management). In the partner system, you can

– list the contents of directories,
– query file attributes, e.g. query the size of a send file,
– modify file attributes, e.g. rename files,
– delete files.
– create, rename and delete directories

openFT for Windows and openFT for Unix systems also offer the option of renaming, creating or deleting directories in openFT partner systems. Partner systems, which support the file management function can also assume the initiative for such requests and access their local system accordingly from the remote system. In both cases, the system in which the initiative has been taken sends a description of the request to the partner system. The partner system executes the request according to its conventions.

If the partner system is a z/OS system, a number of special issues need to be observed. You will find details in the User Guide for openFT for z/OS.
The file management functions are performed via the appropriate protocols (openFT, FTAM or FTP). You can detect differences in the protocols between openFT, FTAM and FTP partner systems by changing the file attributes. Depending on the protocol, and what the partner system supports, you can modify the following attributes of a file.

Attribute	FTAM partner	openFT partner	FTP partner
File name (FILE-NAME/NEW-NAME)	X	X	X
Access rights (ACCESS-MODE)	X	X	
File availability (FILE-AVAILABILITY)	X		
Account for file storage costs (STORAGE-ACCOUNT)	X		
Legal qualification for using a file (LEGAL-QUALIFICATION)	X		
Future file size (FUTURE-FILE-SIZE)	X		

3.7.2 File management in the local system

When using the FTAM functionality, you have the option of assigning special FTAM attributes to file in the local system for communication with FTAM partners (see page 108). The functionality offered by this approach allows you to display and modify FTAM attributes of a file in the local system.

The FTAM attributes exist only in the virtual filestore and primarily valid for file transfer and file management with FTAM partners. In the local system, the operating-system specific setting of the file attributes remains unaltered. This means that This means that files and file attributes can still be modified using commands specific to the operating system. For example, a file can be deleted using a system-specific delete command although the corresponding setting of PERMITTED-ACTION prohibits deletion of the file for FTAM partners.
The following table shows the file management functions in the local system:

FTAM attribute	**display**[1]	**modify**
FILE-NAME *	X	
STORAGE-ACCOUNT	X	
Type of last file usage *	X	
Name of last user of file *	X	
Date and time of last change of file contents	X	
DATA-TYPE	X	X
CHARACTER-SET *	X	X
RECORD-FORMAT *	X	X
Maximum record length (RECORD-SIZE) *	X	X
File availability (FILE-AVAILABILITY) *	X	
Access rights (PERMITTED-ACTIONS) *	X	X
Current file size in bytes (CURRENT-FILE-SIZE) *	X	
Possible file size in bytes (FUTURE-FILE-SIZE)	X	
Legal qualifications (LEGAL-QUALIFICATION)	X	

[1] With local file management, only the FTAM attributes marked * are displayed; for remote file management, all attributes are displayed.

The following FTAM attributes are evaluated for file transfers using the openFT protocol and in part for the FTP protocol:

- Data type (DATA-TYPE)
- Record format (RECORD-FORMAT)
- Maximum record length (RECORD-SIZE)

If the format attributes specified in the file transfer request are not consistent with these FTAM attributes, the request is generally rejected. To avoid this, the FTAM attributes can be deleted in the local file without deleting the file itself.

These FTAM attributes are, however, never set when a file is created using the openFT or FTP protocol.

3.8 Special points for file transfer with FTAM partners

The FTAM functionality allows you to execute file transfer on the basis of ISO protocol ISO8571. The sections below describe special points for “FTAM specialists” with respect to transfer and mapping of FTAM-specific file attributes for file transfer with FTAM partners.

3.8.1 Virtual filestore

Any system that is to enable file transfer using FTAM protocols must make its files available to partner systems in a format that is defined by standard (ISO8571). For this purpose a file’s attributes are mapped from the real filestore onto a virtual filestore and vice versa. The virtual filestore thus has no effect on the attributes of the files in the local system, but has only the tasks of transporting file attributes to the remote FTAM system. In the sections below, the criteria for describing a a file in the virtual filestore are introduced. The format of the virtual filestore is defined by the FTAM standard. Basically, a distinction is made between three different groups of file attributes:

Kernel group

describes the basic attributes of the files. These are specified when the file is created. They include the file name, information relating to the file structure and file contents, and details of agreed file access rights.

Storage group

covers the storage attributes of files. The storage attributes include the file size, the file availability, the date and time of the last read or write access, as well as identification of the user who initiated this in access.

Security group

defines the security attributes for access protection.

Attributes of the kernel group

The attributes in the kernel group are set when the file is created, and contain the basic information on a file:

file name
: contains the file name.

permitted actions
: define which actions can be performed for a certain file:
  – read file (READ-FILE)
  – insert data unit (INSERT-DATA-UNIT)
  – replace (overwrite) file (REPLACE-FILE)
  – extend file (EXTEND-FILE)
  – erase data unit (ERASE-DATA-UNIT)
  – read file attributes (READ-ATTRIBUTES)
  – modify file attributes (CHANGE-ATTRIBUTES)
  – delete file (DELETE-FILE)

  The *permitted actions* also define the method that can be used to access structured files (see also the section “FTAM files” on page 79).

  – forwards (TRAVERSAL)
  – backwards (REVERSE TRAVERSAL)
  – any (RANDOM)

contents type
: Defines the data structure and the method that can be used to access the structured data.

Attributes of the storage group

The attributes of the storage group describe the filestore properties, for example who last accessed the file, the type of access, and when. Some of these properties are automatically modified when the file is read or modified. However, they cannot be modified directly using user commands. You can influence directly modifiable attributes with openFT.

Attribute[1]	**Definition**
storage account *	identifies who is responsible for the file storage costs
date and time of creation	indicates the date and time of creation
date and time of last modification	indicates the date and time of the last modification

Attribute[1]	Definition
date and time of last read access	indicates the date and time of the last read access
date and time of last attribute modification	indicates the date and time of the last attribute modification
identity of creator	identifies the user who created the file
identity of last modifier	identifies the user who last modified the file
identity of last reader	identifies the user who last read the file
identity of last attribute modifier	identifies the user who last modified the file attributes
file availability *	provides information on whether a file is available immediately, or whether it must first be obtained, e.g. from an archive
file size	describes the storage capacity occupied in the actual filestore. A file can thus differ in size in systems that display file types in different ways. Some filestores assign a multiple of a basic unit, e.g. blocks, for file storage. *file size* thus specifies a value that does not correspond to the file size
future file size *	describes the future file size, i.e. possible file size after processing. The initiator can modify the *future file size* value. As soon as the file reaches the specified file size, the responder can increase the value with or without a warning to the initiator. Alternatively, the responder can reject the modification of a value with an appropriate error message.

[1] Attributes marked * are directly modifiable.

Attributes of the security group

The FTAM virtual filestore concept provides a security group for access protection.

Attribute[1]	Definition
access control *	indicates the conditions governing access to files. For example, this may include passwords for various types of access (read, insert, replace, extend), or locks that are used to regulate simultaneous access to a file by different users.
legal qualifications *	specify the legal status of the file and its usage. At present, there is no accepted interpretation of this attribute, i.e. its interpretation depends on the particular partner.

[1] Attributes marked * may be modified directly

3.8.2 Mapping file access rights

This chapter describes how Unix systems's file protection bits are mapped to file management access rights, according to the Siemens openFT protocols and as described in the ISO FTAM standard. It provides information on how to modify and display file access rights using the file management functions. A distinction is made here between requests initiated in the local system (outbound) and those initiated in the remote system (inbound).

3.8.2.1 Outbound requests

You can display and modify the file management access rights for files in the remote system.

Display access rights

The access rights for files in the remote system can be displayed using the FT command *ftshw*. The following file management access rights are displayed:

r (read)	read file
p (replace)	overwrite file
x (extend)	extend file
e (erase)	erase data unit (File Access Data Unit FADU), practical for FTAM partners only
a (rdatt)	read file attributes
c (chatt)	change file attributes
d (delete)	delete file

If openFT is installed in the remote Unix system, the file protection bits *r*, *w*, and *x* are mapped to the file access rights as described in the next section for inbound requests.

For FTAM partners, the more restrictive value for access rights, changeable (access control) or unchangeable (permitted actions), is displayed for the respective FTAM partner, since it is relevant for possible file manipulation.

Modify access rights

You can use the FT command *ftmod* to modify file access rights. The access rights of the receive file can also be set or modified for file transfer requests with FTAM. The individual command descriptions indicate which protection bits can be set and how they are to be set in a remote openFT for Unix system. Access mode options (or combinations of those options) that are not supported are rejected by the file management request, and are ignored by the file transfer request.

3.8.2.2 Inbound requests

Partners in remote systems can display or modify the file management access rights of their own local files.

Display access rights

With a corresponding request from the remote system, openFT for Unix systems maps the local protection bits *r*, *w*, and *x* to the file management access rights as follows:

Access right displayed	**Unix protection bit for the file**	**Unix protection bit for the parent directories**
r (read) read file	r bit	x bit [1]
p (replace) overwrite file	w bit	x bit
x (extend) extend file	w bit	x bit
e (erase) [2] erase data unit	w bit	x bit
a (rdatt) read file attribute		x bit
c (chatt) change file attribute	the request must have the same owner authorization as the file	x bit w bit for the next parent directory
d (delete) delete file	w bit	x bit w bit for the next parent directory

[1]The r bit of the parent directory is not significant.
[2]The attribute is practical for FTAM connections only.

The access right *i* (insert data unit FADU) is not permitted in Unix systems.

The access rights of only one user class (owner, group, other) are displayed. The user class is displayed in accordance with the access authorization for the file management request in the Unix system. If a number of user classes have access authorization, the access rights for the highest user class are displayed (e.g. owner access rights before group access rights).

Furthermore, local Unix system rules apply to file access. Thus, for example, the x bit must be set for all parent directories.

Modify access rights

The following table shows the options available in Unix systems for modifying file protection bits:

File management access rights	Unix file protection bits	Function
rpxeacd	rw[1]	read-write
rac	r-[1]	read-only
pxeacd	-w[1]	write-only
ac	--[1]	none

[1]The x bit is not changed by the respective openFT command from the Unix system. From Windows-PCs, even the attributes of remote directories can be changed. In this case, even the x bit is set by rpxeacd (=@rw).

The openFT protocols and FTAM only recognize two options for access rights, namely ‘set’ and ‘not set’. This means that when entering access rights, it is necessary to specify whether or not the access right is set. These protocols do not provide the option of leaving access rights unchanged.

To enable file access rights to be modified, the file management access rights *a* and *c* must always be specified; otherwise, the remote request is rejected.If the w protection bit is to be set for a file, the file access rights *pxed* must also be set, since all these values are mapped to the w file protection bit. All other combinations of file access rights cause the remote request to be rejected.

Only the file owner can modify the access rights of a particular file. Access rights set by the owner can only be modified by the user class ’owner’. However, owner, group, and other user classes can delete access authorizations.

3.8.3 Mapping FTAM attributes to the real file system

This section describes the way in which the FTAM implements the virtual filestore, and the mechanisms used for mapping virtual and real filestores in Unix systems.

Some FTAM attributes are mapped to the attributes available in Unix systems, and others to the so-called “FTAM catalog”.

The FTAM catalog is used to extend the file attributes available in Unix systems. It is only relevant for access using FTAM. This means that a file can be deleted using the shell command *rm*, even if the *permitted actions* parameter from the FTAM catalog does not permit this for an FTAM partner. This may result in inconsistencies between the FTAM catalog and the real file system. These inconsistencies are detected automatically when openFT for Unix systems s started and the corresponding entries are deleted from the FTAM catalog.

Entries in the FTAM catalog are created using inbound file management requests or a file transfer request, or by modifying the local FTAM attributes. When the file is deleted from the remote system, the appropriate entry in the FTAM catalog is also removed.

It is important to remember that a file identified as a text file in the FTAM catalog, for example, cannot be transferred as a binary file, nor can it be extended by binary data.

The FTAM attributes of a file that are stored in the FTAM catalog are not visible to pointers to the file (such as symbolic links).

3.8.3.1 Inbound mapping of FTAM attributes

The following table shows how FTAM attributes are mapped to the real Unix file system.

Attribute group	FTAM attributes	Mapping in the Unix system (inbound receive)	Modify FTAM attributes
Kernel group	permitted actions READ-FILE INSERT-DATA-UNIT REPLACE-FILE EXTEND-FILE ERASE-DATA-UNIT READ-ATTRIBUTES CHANGE-ATTRIBUTES DELETE-FILE	FTAM catalog	permitted locally[1]
	universal class number GRAPHIC GENERAL IA5 VISIBLE	FTAM catalog	permitted locally [1]
	string significance VARIABLE FIXED not significant	FTAM catalog	permitted locally [1]
	maximum string length	FTAM catalog	permitted locally [1]
	document type FTAM1 FTAM3	FTAM catalog	permitted locally [1]
Storage group	file availability IMMEDIATE DEFERRED	FTAM catalog	inbound permitted
	future file size	is ignored	not permitted
	storage account	is ignored	not permitted

Attribute group	FTAM attributes	Mapping in the Unix system (inbound receive)	Modify FTAM attributes
Security group	ActionList (of 1ACE)		
	READ-FILE	r	inbound permitted
	INSERT-DATA-UNIT	not permitted	not permitted
	REPLACE-FILE	w	inbound permitted
	EXTEND-FILE	w	inbound permitted
	ERASE-DATA-UNIT	w	inbound permitted
	READ-ATTRIBUTES	x dir	inbound permitted [2]
	CHANGE-ATTRIBUTES	w dir+owner	inbound permitted [2]
	DELETE-FILE	w + wdir	inbound permitted
	LEGAL-QUALIFICATION	is ignored	not permitted

[1] A local modification of the FTAM attribute is possible with the *ftmodf* function.
[2] The value must always be sent, but may not be changed.

The following file attributes are derived from the current Unix file attributes:
– file name
– file size
– identity of creator
– date and time of last read access
– date and time of last attribute modification
– date and time of last modification
– access control

Other attributes are only partially supported by openFT for Unix systems. As the responder, openFT for Unix systems does not return any value for the following file attributes (*no value available*):
– identity of last modifier
– identity of last reader
– identity of last attribute modifier
– storage account
– legal qualification

In Unix systems, the FTAM protocol parameter *filestore password* is mapped to the password of the of the login name concerned.

3.8.3.2 Inbound mapping the document type

The following tables provide information on mapping the *document type* during file transfer. A distinction is made here between openFT for as the receiving system and openFT for as the sending system.

Mapping of the document type for Inbound Receive (FTAM --> Unix system)

FTAM (virtual filestore in the remote system)			**Unix receive file**
document type	universal class	string significance	
FTAM-1	25 - GraphicString	variable/fixed	text file
FTAM-1	26 - VisibleString	variable/fixed	text file
FTAM-1	27 - GeneralString	not significant	text file
FTAM-1	22 - IA5String	not significant	text file
FTAM-3	----	not significant	unstructured binary file
FTAM-3	----	variable	record-structured binary file
FTAM-3	----	fix	binary file with fixed record structure

No provision in made for transfer of FTAM-3 files with variable and fixed *string significance* in the functional standard ENV 41204. openFT for Unix systems provides additional support for this function, since the file format corresponds to the user format in Unix systems .

Mapping of the document type for Inbound Send (FTAM <-- Unix system)

FTAM (specifications in request and/or entries in the FTAM catalog in the local system)			**Unix send file**
document type	universal class	string significance	
not specified	not specified	not specified	text file
FTAM-1	not specified	not specified	text file
FTAM-1	25 - GraphicString	variable/fixed	text file
FTAM-1	26 - VisibleString	variable/fixed	text file
FTAM-1	27 - GeneralString	not significant	text file
FTAM-1	22 - IA5String	not significant	text file
FTAM-3	----	not specified	unstructured binary file
FTAM-3	----	not significant	unstructured binary file
FTAM-3	----v	variable	record-structured binary file
FTAM-3	----	fix	binary file with fixed record structure

If there is an entry for the Unix send file in the FTAM catalog, the file format specifications in the request must correspond to this entry. Otherwise, files inconsistencies may occur and file transfer requests involving the particular file may be aborted.

If there are no specifications in the request, the entries in the FTAM catalog apply.

3.8.3.3 Access protection

As explained in the section “Virtual filestore” on page 108, openFT supports the security die Security group of the virtual filestore. This provides an effective protection mechanism against unauthorized access to files.

For access authorization to the virtual filestore of a system you need the FTAM protocol parameters *initiator identity* and *filestore password*. openFT for Unix systems maps these parameters to the login name and its password in Unix.

For file transfers with FTAM partners it is also possible to use the FTAC functions for extended protection against unauthorized forms of access. If an admission profile in Unix systems is to be addressed by an FTAM partner, then the transfer admission for the profile concerned must be supplied in the protocol parameter *initiator identity*. The parameters *filestore password* and *account* must not be specified. Apart from this, the rules of the FTAC functions described in this manual apply here (e.g. referencing a file that has been predefined in the admission profile either with the specification *NOT-SPECIFIED for the file name, or by omitting the file name, etc.).

3.8.3.4 Outbound mapping of the document type

If openFT for Unix systems is the initiator, the FT user can use the file type specification (options *-t*, *-u*, *-b* in *ft* and *ncopy* command) to specify in the request whether text or binary data is to be transferred. There is no attribute for binary or text data in the real store on the Unix system.

The following tables provide information on mapping the *document type* during file transfer. A distinction is made here between openFT as the receiving system and the sending system.

Outbound Sending (Unix system --> FTAM)

Unix system	**FTAM (virtual filestore in the remote system)**		
File type	document type	universal class	string significance
Text (-t)	FTAM-1	25 - GraphicString	variable [1)]
User format (-u)	FTAM-3	----	variable [1)]
Binary (-b)	FTAM-3	----	not significant [1)]
Binary + record length (-b -r=max record length)	FTAM-3	----	fixed

[1)] If one of the options *-t*, *-u*, or *-b* are specified and an entry for the send file on the Unix system exists in the FTAM catalog extension, this entry must correspond to the entries in the above table.

If the FT user does not specify a file type in the request, the entries in the FTAM catalog are used. If there is no entry in the FTAM catalog, FTAM1, GraphicString, and variable are used.

No provision is made for transfer of FTAM-3 files with variable *string significance* in the functional standard ENV 41204. openFT for Unix systems provides additional support for this function.

Outbound Receive (Unix system <-- FTAM)

For outbound receive, the type of the Unix receive file depends on whether and which file type, if any, was specified in the FT request. The following cases must be differentiated here.

1. No file is specified in the request

FTAM (virtual filestore in the remote system)			**Receive file on the Unix system**
document type	universal class	string significance	
FTAM-1	25 - GraphicString	variable/fixed	text file
FTAM-1	26 - VisibleString	variable/fixed	text file
FTAM-1	27 - GeneralString	not significant	text file
FTAM-1	22 - IA5String	not significant	text file
FTAM-3	---	not significant	unstructured binary file
FTAM-3	---	variable	record-structured binary file
FTAM-3	----	fix	binary file with fixed record structure

2. *-t* option resp. *Text Format* specified for file type in request

FTAM (virtual filestore in the remote system)			**Receive file on the Unix system**
document type	universal class	string significance	
FTAM-1	25 - GraphicString	variable/fixed	text file
FTAM-1	26 - VisibleString	variable/fixed	text file
FTAM-1	27 - GeneralString	not significant	text file
FTAM-1	22 - IA5String	not significant	text file

3. *-u* option resp. *User format* specified for file type in the request

FTAM (virtual filestore in the remote system)			**Receive file on the Unix system**
document type	universal class	string significance	
FTAM-3	---	variable	record-structured binary file
FTAM-3	----	fix	binary file with fixed record structure

4. -*b* option resp. *Binary* specified for file type in the request

FTAM (virtual filestore in the remote system)			**Receive file on the Unix system**
document type	universal class	string significance	
FTAM-3	---	not significant	unstructured binary file

5. -*b* and -*r* (max. record length) options resp. *Binary* + *Maximum Record Length* specified for file type in the request

FTAM (virtual filestore in the remote system)			**Receive file on the Unix system**
document type	universal class	string significance	
FTAM-3	----	fix	binary file with fixed record structure

3.8.4 FTAM diagnostic codes as per ISO 8571-3

The following excerpt from ISO FTAM standard ISO 8571-3 describes the possible diagnostic codes that can appear in the DIAGCODE column or in the messages 2093 or 2215 as $NUMMER when displaying the request queue for requests to FTAM partners (see the section “Reason codes of the logging function” on page 288):

Identifier	Reason
0	No reason
1	Responder error (unspecific)
2	System shutdown
3	FTAM management problem (unspecific)
4	FTAM management, bad account
5	FTAM management, security not passed
6	Delay may be encountered
7	Initiator error (unspecific)
8	Subsequent error
9	Temporal insufficiency of resources
10	Access request violates VFS security
11	Access request violates local security
1000	Conflicting parameter values
1001	Unsupported parameter values
1002	Mandatory parameter not set
1003	Unsupported parameter
1004	Duplicated parameter
1005	Illegal parameter type
1006	Unsupported parameter types
1007	FTAM protocol error (unspecific)
1008	FTAM protocol error, procedure error
1009	FTAM protocol error, functional unit error
1010	FTAM protocol error, corruption error
1011	Lower layer failure

Identifier	Reason
1012	Lower layer addressing error
1013	Timeout
1014	System shutdown
1015	Illegal grouping sequence
1016	Grouping threshold violation
1017	Specific PDU request inconsistent with the current requested access
2000	Association with user not allowed
2001	(not assigned)
2002	Unsupported service class
2003	Unsupported functional unit
2004	Attribute group error (unspecific)
2005	Attribute group not supported
2006	Attribute group not allowed
2007	Bad account
2008	Association management (unspecific)
2009	Association management - bad address
2010	Association management - bad account
2011	Checkpoint window error - too large
2012	Checkpoint window error - too small
2013	Checkpoint window error - unsupported
2014	Communications QoS not supported
2015	Initiator identity unacceptable
2016	Context management refused
2017	Rollback not available
2018	Contents type list cut by responder
2019	Contents type list by Presentation service
2020	Invalid filestore password
2021	Incompatible service classes
3000	Filename not found

Identifier	Reason
3001	Selection attributes not matched
3002	Initial attributes not possible
3003	Bad attribute name
3004	Non-existent file
3005	File already exists
3006	File cannot be created
3007	File can not be deleted
3008	Concurrence control not available
3009	Concurrence control not supported
3010	Concurrence control not possible
3011	More restrictive lock
3012	File busy
3013	File not available
3014	Access control not available
3015	Access control not supported
3016	Access control inconsistent
3017	Filename truncated
3018	Initial attributes altered
3019	Bad account
3020	Override selected existing file
3021	Override deleted and recreated file with old attributes
3022	Create override deleted and recreate file with new attributes
3023	Create override - not possible
3024	Ambiguous file specification
3025	Invalid create password
3026	Invalid delete password on override
3027	Bad attribute value
3028	Requested access violates permitted actions
3029	Functional unit not available for requested access

Identifier	Reason
3030	File created but not selected
4000	Attribute non - existent
4001	Attribute cannot be read
4002	Attribute cannot be changed
4003	Attribute not supported
4004	Bad attribute name
4005	Bad attribute value
4006	Attribute partially supported
4007	Additional set attribute value not distinct
5000	Bad FADU (unspecific)
5001	Bad FADU - size error
5002	Bad FADU - type error
5003	Bad FADU - poorly specified
5004	Bad FADU - bad location
5004	FADU does not exist
5006	FADU not available (unspecific)
5007	FADU not available for reading
5008	FADU not available for writing
5009	FADU not available for location
5010	FADU not available for erasure
5011	FADU cannot be inserted
5012	FADU cannot be replaced
5013	FADU cannot be located
5014	Bad data element type
5015	Operation not available
5016	Operation not supported
5017	Operation inconsistent
5018	Concurrence control not available
5019	Concurrence control not supported

Identifier	Reason
5020	Concurrence control inconsistent
5021	Processing mode not available
5022	Processing mode not supported
5023	Processing mode inconsistent
5024	Access context not available
5025	Access context not supported
5026	Bad write (unspecific)
5027	Bad read (unspecific)
5028	Local failure (unspecific)
5029	Local failure - filespace exhausted
5030	Local failure - data corrupted
5031	Local failure - device failure
5032	Future file size exceeded
5034	Future file size increased
5035	Functional unit invalid in processing mode
5036	Contents type inconsistent
5037	Contents type simplified
5038	Duplicate FADU name
5039	Damage to select/open regime
5040	FADU locking not available on file
5041	FADU locked by another user
6000	Bad checkpoint (unspecific)
6001	Activity not unique
6002	Checkpoint outside window
6003	Activity no longer exists
6004	Activity not recognized
6005	No docket
6006	Corrupt docket
6007	File waiting restart

Identifier	Reason
6008	Bad recovery point
6009	Non-existent recovery point
6010	Recovery mode not available
6011	Recovery mode inconsistent
6012	Recovery mode reduced
6013	Access control not available
6014	Access control not supported
6015	Access control inconsistent
6016	Contents type inconsistent
6017	Contents type simplified

3.8.5 Addressing via Application Entity Title (AET)

In the OSI world, communication partners are represented by application entities. An application entity is an addressable entity in Layer 7 of the OSI Reference Model (Application Layer). Such an application entity is the access point of an FTAM application, for example, via which an OSI-TP communication partner can connect to the FTAM application. In the OSI-TP standard, every application entity is assigned to an application entity title, via which the application entity can be addressed uniquely in the OSI network.

Two forms of AET are defined in the ISO Standard, the Directory Form and the Object Identifier Form. openFT-FTAM for BS2000 by default sends a "Nil-Application Entity Title". The FTAM functions of openFT for Unix systems and openFT for Windows support the Object Identifier Form of the AET. An AET comprises two parts:

– Application Process Title (APT)
– Application Entity Qualifier (AEQ).

When transmitting with the FTAM protocol, openFT sends a Nil Application Entity Title as a calling or called Application Entity Title by default. This behavior can be modified if desired (see the description of the *ftmodo -ae* command and in the online help).

The Nil AET is:
`1.3.9999.1.7`

Addressing FTAM partners with AET

If a called AET is to differ from the "Nil Application Entity Title" then it must be specified in the partner list on instance identification (command: *ftaddptn -id*).

The specification has the following syntax:

n1.n2[.n3] [.n10][..m]

n1.n2[.n3] [.n10]
: specifies the *application process title*, between two and ten numbers separated by a period (.). The range and the meaning of the numbers are explained below.

[..m] specifies the *application entity qualifier*, range of *m* see below. The two periods are mandatory if a AEQ is specified.

Example

A FTAM partner on computer *daisy2* with APT=*1.0.56.881.4* and AEQ=*785* is to be entered in the partner list under the name *daisyftm*. To do this, enter the following command:

```
ftaddptn daisyftm -pa=ftam://daisy2 -id=1.0.56.881.4..785
```

Application Process Title (APT)

The APT used to identify the application. The APT should be unique worldwide in accordance with the OSI Standard. For this reason, it should be issued and registered by a Standardization Committee).

An APT in Object Identifier Form is consists of up to 10 components:
`(component1,component2,...,component10)`

The values for component1 to component10 are partially standardized. In this context, a symbolic name was assigned to several numbers. The range of values for component2 depends on the value of component1. The following table shows the symbolic names and the value ranges of the functions supported by FTAM:

component1	**component2**	**component3 to component10**
0: CCITT	0: RECOMMENDATION 1: QUESTION 2: ADMINISTRATION 3: NETWORK-OPERATOR (permissible values: 0 - 39)	Permissible values: 0 - 67 108 863
1: ISO	0: STANDARD 1: REGISTRATION-AUTHORITY 2: MEMBER-BODY 3: IDENTIFIED-ORGANIZATION (permissible values: 0 - 39)	Permissible values: 0 - 67 108 863
2:JOINT-ISO-CCITT	Permissible values: 0 - 67 108 863	Permissible values: 0 - 67 108 863

The APT which you specify need not be stipulated by a standardization committee, i.e. you may stipulate your own APT. It must satisfy the following two conditions:

– it must be unique throughout the network
– it must be made up of values that are permissible according to the table above

A remote partner that requests AETs must know this APT in order to set up a connection.

Application Entity Qualifier (AEQ)

The AEQ identifies an access point within an application. You can assign AEQs to the access points of an application only if you have assigned an APT to that application. It is assigned by the operator of the application.

The AEQ is a positive whole number between 0 and 67108863.

You must not use the same AEQ more than once within an application, i.e there must never be two access points with the same AEQ in one application. However, you do not have to assign all the access points in an application to an AEQ.

4 Working with openFT

This chapter describes how you can work with openFT by various methods using the graphical interface, the menu system, openFT commands, and the program interface.

4.1 The openFT Explorer for X Window

If you are working with an X terminal under the X Window interface, you can use the functions of openFT via the graphical user interface - the openFT Explorer.

Starting and exiting the openFT Explorer

The openFT Explorer is called by entering the command *openFT* from the shell. Before you call openFT, make sure that the shell variable *DISPLAY* has been set, since this variable determines on which terminal the openFT Explorer is to be displayed.

For example, if you want the outputs to be displayed on a computer with the IP address 47.11.08.15, you must first invoke the following command before the call:

```
DISPLAY="47.11.08.15:0"; export DISPLAY
```

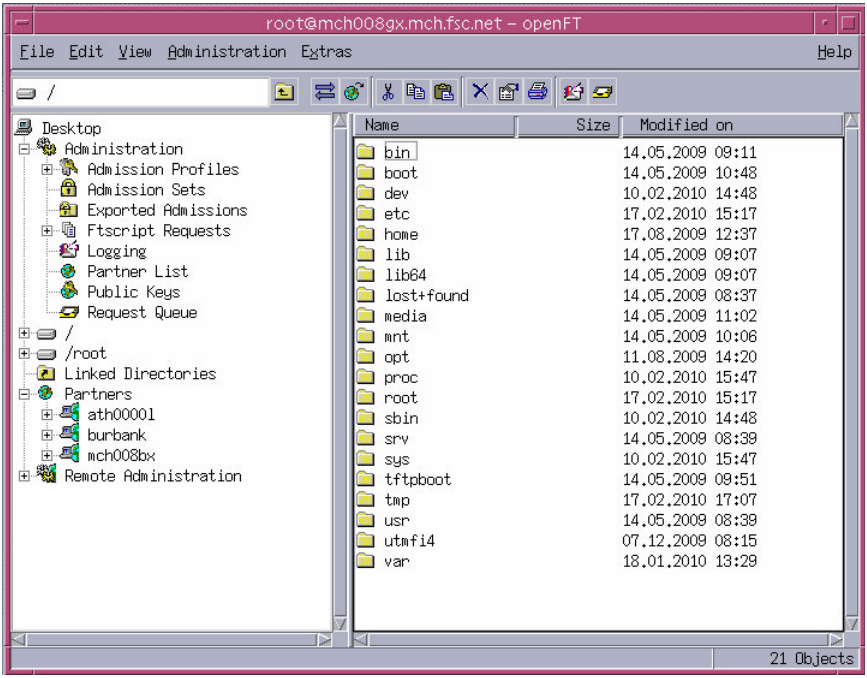
If you are calling the openFT Explorer the first time under your login name, the online help window is opened in addition to the openFT window with a description of the graphical interface. You can switch back and forth between these two windows at any time.

You can close and exit the openFT Explorer by either clicking the Close button on these windows or by the menu entry *File - Exit*.

Operating the openFT Explorer

Working with the openFT Explorer is analogous to working with Microsoft´s Windows Explorer. The object directories (i.e. the folders) appear in the left pane of the window, and the objects of the selected folder appear on the right.

Following the first call, the folders appear in a structure similar to the one shown in the example below:



Transfers are accomplished by the drag & drop technique, i.e., by clicking the desired object, holding down the mouse button, dragging it to a new location, and releasing the mouse button.

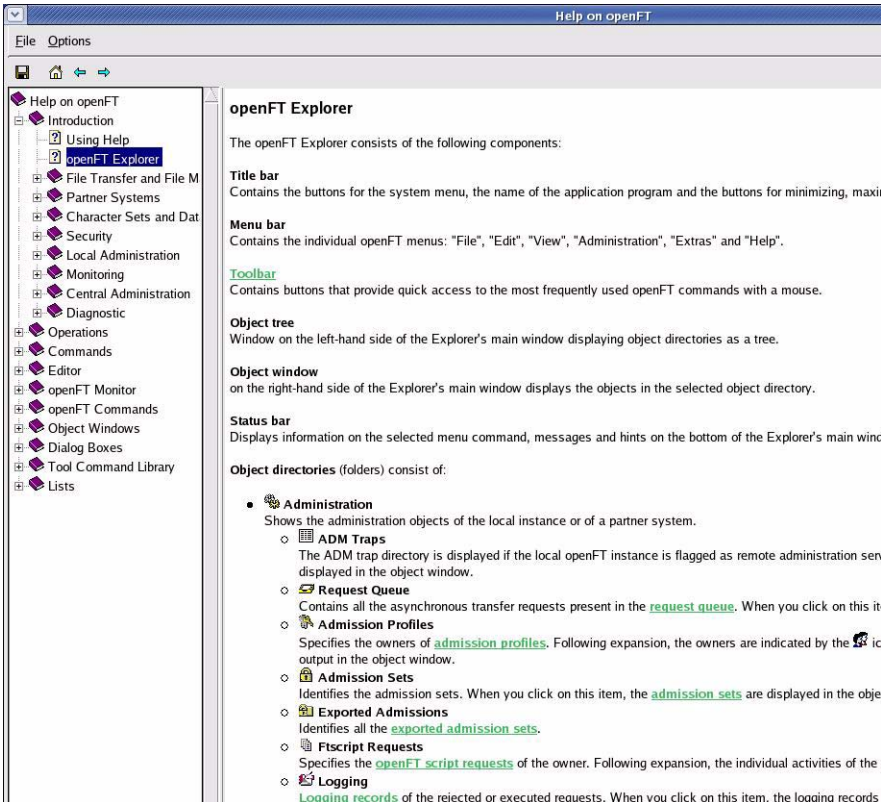
The openFT Explorer uses the clipboard for temporary storage under X Window. If you are working under an X Window emulation and want to combine the clipboards of the emulation and your operating system (e.g. Windows), you will need to set the preferences of the emulation accordingly (see the documentation of your X Window emulation for details).

Before starting the openFT Explorer, make sure that the Num Lock key has not been pressed. On Linux systems, the Num Lock key acts as an Alt lock key. This can cause problems navigating in the object tree in the openFT Explorer. For example, it is not possible to view file attributes, directories, FTAC profiles or log records, as the pressed Num Lock key causes mouse events to be changed (a click becomes Alt + click, and a double-click becomes Alt + double-click).

For further details, please refer to the online help supplied with the openFT Explorer.

Online help on the openFT Explorer

The online help for the openFT Explorer is automatically called when you start it for the first time under your login name (or user ID). You can then call for online help at any time by clicking *Help* - *Contents* on the menu bar. Depending on what you select, you will receive a help window similar to the one below:


Help on openFT
File Options
Help on openFT
Introduction
Using Help
openFT Explorer
File Transfer and File M
Partner Systems
Character Sets and Dat
Security
Local Administration
Monitoring
Central Administration
Diagnostic
Operations
Commands
Editor
openFT Monitor
openFT Commands
Object Windows
Dialog Boxes
Tool Command Library
Lists
openFT Explorer
The openFT Explorer consists of the following components:
Title bar
Contains the buttons for the system menu, the name of the application program and the buttons for minimizing, maxi
Menu bar
Contains the individual openFT menus: "File", "Edit", "View", "Administration", "Extras" and "Help".
Toolbar
Contains buttons that provide quick access to the most frequently used openFT commands with a mouse.
Object tree
Window on the left-hand side of the Explorer's main window displaying object directories as a tree.
Object window
on the right-hand side of the Explorer's main window displays the objects in the selected object directory.
Status bar
Displays information on the selected menu command, messages and hints on the bottom of the Explorer's main winc
Object directories (folders) consist of:
Administration
Shows the administration objects of the local instance or of a partner system.
ADM Traps
The ADM trap directory is displayed if the local openFT instance is flagged as remote administration sen
displayed in the object window.
Request Queue
Contains all the asynchronous transfer requests present in the request queue. When you click on this it
Admission Profiles
Specifies the owners of admission profiles. Following expansion, the owners are indicated by the ic
output in the object window.
Admission Sets
Identifies the admission sets. When you click on this item, the admission sets are displayed in the obje
Exported Admissions
Identifies all the exported admission sets.
Ftscript Requests
Specifies the openFT script requests of the owner. Following expansion, the individual activities of the
Logging
Logging records of the rejected or executed requests. When you click on this item, the logging records

In addition, most dialog boxes provide context-sensitive help that you can call up by pressing the *Help* button or the F1 key.

Configuration files of the openFT Explorer

The openFT Explorer generates three configuration files: *.openFTxcfg, .openFTrc* and *.openFTeditrc*, which may automatically change during the course of a session. By default, only the owner of these files is allowed read or write access to them *(-rw-------)*.

You must not change the contents of these files manually.

The file *.openFTxcfg* contains the list of partner systems entered via the graphical interface and the list of directory links created by the user. This file may also be used to save transfer admissions and other data, e.g. for remote administration.

The settings for the openFT Explorer are saved in the file *.openFTrc*.

The settings for the *ftedit* are saved in the file *.openFTeditrc*.

4.2 The openFT-Script interface

openFT-Script provides you with a script language in XML notation. This comprises the openFT functions familiar from the command or C interface as well as offering additional context management and control functions.

The XML statements in an openFT-Script request are stored in a text file. These files can be edited with a text editor or any desired XML tools. No compiler is required. The J2SE™ Runtime Environment 5.0 (JRE 5.0) or higher is required for execution.

An openFT-Script request is started using the *ftscript* command. In addition, the openFT-Script interface offers further commands for the administration of openFT-Script runs, see page 355. openFT-Script requests can also be monitored and cancelled in the **Ftscript Requests** object directory in the openFT Explorer.

A detailed description of the XML interface can be found in the manual "openFT-Script Interface".

4.3 The openFT commands

openFT can also be operated and managed via commands, which means that you can also create shell scripts for tasks to be performed using openFT. The commands which are relevant for users are described in this manual as of page 139. Commands for administrators can be found in the System Administrator Guide.

Besides the openFT commands, even the BSFT commands and some CMX commands are described in these manuals.

More details on the layout of the command descriptions can be found in the corresponding chapters.

Help on the commands (manpages)

For each command, there is also a corresponding man page, which you can call from the command line by using the UNIX command *man* together with the name of a command as its argument (e.g. *man ft*).

4.4 Program interface

Using the program interface of openFT, even user-defined C or Java applications can access its functionality. A wide range of functions are available for this purpose. Further details on the program interface are presented in the chapter “Program interface” on page 373.

5 openFT commands for the user

This chapter contains a functional description of openFT commands, as well as detailed descriptions of the individual commands. The functional command description provides you with a quick overview of which commands are available for which tasks. This is followed by an explanation of the notational conventions used in the command descriptions. Finally, the commands are described in alphabetical order.

The commands for the openFT script interface are described in chapter “openFT-Script Commands” on page 355 as well as in the "openFT Script Interface" manual.

5.1 Overview of the commands

The following overview shows a list of all commands for users arranged according to the various tasks.

A graphics-capable terminal is required for commands marked [g].

File transfer and request queue managing

ncopy	Issue synchronous file transfer request
ft	Issue asynchronous file transfer request
ftcanr	Cancel asynchronous file transfer requests
ftmodr	Change the order of the requests in the request queue
ftshwr	Display the properties and statuses of requests

Remote command execution

ftexec	Execute operating system commands in remote system

File management

ftcredir	Create remote directories
ftshw	Display attributes of one or more files in the remote system

ftshwf	Display the FTAM attributes of a local file
ftmod	Modify file attributes in a remote system
ftmoddir	Modify the attributes of remote directories
ftmodf	Modify the FTAM attributes of a local file
ftdel	Delete a file in a remote system
ftdeldir	Delete remote directories

Logging

ftshwl	Display log records
fthelp	Display information on the reason codes in the log records

FTAC function

ftcrep	Create FT profile
ftshwp	Display FT profile
ftmodp	Modify FT profile
ftdelp	Delete FT profile
ftshwa	Display admission set
ftmoda	Modify admission set

Administer instances

ftseti	Set an instance
ftshwi	Output information on instances

Display measurement data

ftshwm	Display measurement data of the openFT operation
ftmonitor [g]	Display measurement data of the openFT operation on openFT Monitor

Output of general information and miscellaneous commands

ftinfo	Output information about the openFT system
ftshwo	Display operating parameters
ftshwptn	Display partner properties
ftedit [g]	Load local or remote files in the openFT editor
ftmsg [g]	Output message box on a graphical display

[g] A graphics-capable terminal is required for this command

5.2 Notational conventions

The command syntax essentially corresponds to the output that you get when you specify the command with *-h* option. The following conventions have been used for syntax diagrams:

< > angle brackets are used for parameters which you may replace with current values. You must not specify the angle brackets < > and the permissible value ranges.

[ ] enclose optional entries. The effect on the function of the command is described for the individual parameters.

␣ stands for at least one blank that must be inserted between the various entries.

| stands for alternatives. You may specify only one of the values indicated.

Bold typeface
This is used in the "Description" sections for individual characters or strings that must be specified in exactly the form given, e.g. options or values.
In running text, these are then shown in *italics*.

Lengths and characters sets

The values which you use for parameters in the commands must observe certain restrictions on length and on the characters available:

file name
you can specify an absolute or relative file name.
The file name specified in the local and remote systems may have a maximum length of 512 characters based on the length of the absolute path name. Please note that although long file names can be specified at the openFT interfaces, not all platforms support this maximum length. For example Unix systems permit up to 512 characters whereas Windows systems only permit 256 characters.

If the file name contains blanks, they must be set in double quotes ("), e.g. "file name".

date
numeric; exactly 8 characters in the form `yyyymmdd` with:
`yyyy` for year, `mm` for month and `dd` for day

Note that for all date entries, you may only specify values up to and including 20380119 (January 19, 2038)

user ID
: User ID for accessing the required system, maximum 64 characters + 3 characters for hexadecimal format (X' '). The maximum length is system-dependent:
In Unix systems, a maximum of 32 characters with first 8 characters being unique; in Windows systems, a maximum of 36 characters.

command
: up to 1000 characters (exception: *ftadm*); for follow-up processing commands, the commands for success and failure must not be longer than 1000 characters in total.

partner
: Name of the partner system in the partner list (1 to 8 characters) or address of the partner system (maximum 200 characters). The address of the partner system is to be specified in the following form:

  [protocol://]host[:[port].[tsel].[ssel].[psel]]

  For further details see section "Defining the partner computer" on page 90.

profile name
: alphanumeric (a..z, A..Z, 0..9), up to 8 characters.

transfer admission
: the transfer admission usually consists of printing characters and may not start with a hyphen, minimum 8 characters, maximum 67 characters (in Unix systems, maximum 32 characters). If a transfer admission consists of non-printing characters then it must be specified in hexadecimal format in the form x\'...\' or X\'...\' .

Special characters

Special characters in the entries for *file name, file name-prefix*, *transfer admission*, *user ID*, *account*, *password*, *follow-up processing* (see notes on the commands) must be escaped using a backslash (\). Here, you must differentiate between special characters for file transfer and special characters on a Unix based operating system, and escape the special characters accordingly.

Note that the entries for command strings, file names and free text must be enclosed in single quotes (') or double quotes (").

If the entry for follow-up processing also contains single quotes (’), it is recommended to enclose the entire entry in double quotes ("). The single quotes in the follow-up processing command (e.g. single quotes in a BS2000 password) can then be written as expected in the partner system (such as BS2000).

Example

The account number 1111111,00000000,88888888 is specified in the transfer admission. The comma is a special character that enables file transfer separating the elements of the triple *user ID*, *account* and *password*, and must therefore be escaped with a backslash (\). This backslash is also a special character for the shell, and must therefore also be escaped. The entry then appears as follows:

"1111111\\,00000000\\,88888888"

Sequence of entries

The **sequence** of entries in the command is arbitrary.
Exceptions to this are for the entries for

– the source and destination of a request (e.g. local and remote file name, partner name,...)
– the authorization to access the remote system, i.e., the transfer admission or the system login.

Continuation lines

When there is a large number of parameters, openFT commands can be very long. If you want to use the keyboard to enter commands that are longer than 256 characters, you will need to work with continuation lines. You can obtain these by entering the sequence \ ↵.

5.3 Output in CSV format

For some Show commands, openFT offers output in CSV format. CSV (**C**omma **S**eparated **V**alues) is a popular format in the PC environment in which tabular data is defined by lines. Output in CSV format is offered for the following commands:

– ftshw
– ftshwa
– ftshwl
– fshwm
– ftshwo
– ftshwp
– ftshwptn
– ftshwr

Output in CSV format is also possible for the openFT-Script commands *ftshwact* and *ftshws*, see "openFT-Script Interface" manual.

Many programs such as spreadsheets, databases, etc., can import data in CSV format. This means that you can use the processing and presentation features of such programs on the data output by the above commands.

The output fields are described in the appendix starting on page 431.

Every record is output as a line, and each record contains information on an object. If data is present, the first line always contains the header with the field names of each of the columns. **Only the field names are guaranteed, not the order of fields in a record**. In other words, the order of fields is determined by the order of the field names in the header line. Fields within an output line are separated by semicolons (;).

The following data types are differentiated in the output:

Number

String
: Since the ";" (semicolon) character has a special meaning in the CSV output as a field separator, a text containing a ";" is enclosed within double quotes. This also applies to the other special characters such as the newline character.
Keywords are never enclosed within double quotes and **always** begin with the character "*" (asterisk).

Date

Date and time are always output in the format `yyyy-mm-dd hh:mm:ss`; a date alone is output in the format `yyyy-mm-dd`.

One example of a possible evaluation procedure is supplied as a reference template in the Microsoft Excel format in the file */opt/openFT/samples/ftaccnt.xlt* . The template evaluates a CSV log file by means of an automatically running macro. The result shows the number of inbound and outbound requests and the Kilobytes transferred in each case for all users.

5.4 ft - Asynchronous file transfer

The *ft* command is used to issue asynchronous file transfer requests for sending a file to a remote system or for fetching a file from a remote system or for executing an operating system command in the local or remote system. Once openFT has stored the request in the request queue, your user process will be available again. openFT performs the actual transfer operation asynchronously to your user process at the earliest opportunity or at a time you specify, provided resources are free and the partner is available.

openFT acknowledges receipt of the request by default, with the output of the following message on the screen (*stderr*) of the user who issued the request

`ft: Request` *request ID* `accepted.`

request ID
: is replaced by the transfer identification of the transfer request.

After acknowledgment of the request, the user process continues to run. If you want, you can use the *-m* option to tell openFT to send a result notification to the initiator's mail box if the request is processed successfully and/or unsuccessfully.

If openFT rejects your request, an error message will be displayed explaining why it was rejected (see chapter "Messages" on page 377).

The maximum number of requests that can be stored in the request queue is specified in the operating parameters. You can raise the default value of 2000 up to a maximum of 32000 (see the *ftmodo* command in the System Administrator Guide). Any further requests are rejected.

You can also obtain the result of an *ft* request by using the log function (see section "ftshwl - Display log records" on page 272).

A number of special issues and restrictions apply for transfer requests with FTP partners. For details, see section "FTP partners" on page 33.

Only one file can be fetched from a remote system for each *ft* command. If you want to fetch several files asynchronously, use the *ft_mget* command. See the section "ft_mget - Fetching multiple files" on page 423.

Format

```
ft  -h |
        [ -t | -u | -b ] [ -x ]
        [ -o | -e | -n ]
        [ -k | -z ][ -c ][ -N ][ -S ][ -m=n | -m=f | -m=a ]
        [ <file name 1..512>    <partner 1..200>![<file name 1..512> ] ]  |
        [ <partner 1..200>![<file name 1..512>]  <file name 1..512> ]
        [ <transfer admission 8..67> | @n | @d |
          <user ID 1..67>[,[<account 1..64>][,[<password 1..64>]]] ]
        [ -p=[<password 1..64>] ][ -di ]
        [ -lc=<CCS name 1..8> ][ -rc=<CCS name 1..8> ]
        [ -ls=<follow-up proc 1..1000> ][ -lf=<follow-up proc 1..1000> ]
        [ -rs=<follow-up proc 1..1000> ][ -rf=<follow-up proc 1..1000> ]
        [ -r=v[<1..32767>] | -r=f[<1..32767>] | -r=u[<1..32767>] |
          -r=<1..32767> ]
        [ -tff=b | -tff=s ][ -trf=u ]
        [ -tb=n | -tb=f | -tb=a ]
        [ -av=i | -av=d ][ -ac=<new account 1..64> ]
        [ -am=[r][i][p][x][e][a][c][d] | -am=@rw | -am=@ro ]
        [ -lq=<legal qualification 1..80> ]
        [ -cp=[<password 1..64>] ][ -pr=n | -pr=l ]
        [ -sd=yyyymmdd | +<start date 0..ddddd> ]
        [ -st=[+]<start time hhmm> ]
        [ -cd=yyyymmdd | +<cancel date 0..ddddd> ]
        [ -ct=[+]<cancel time hhmm> ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

[**-t** | **-u** | **-b**] [**-x**]
Identifies the type of file in the local operating system.

If you send a file to an FTAM partner without specifying a file type, the file type is determined by the structure entries of the send file. The structure entries can be displayed by outputting the local FT attributes (*ftshwf file name -l*). If there are no structure entries, the default value is *-t*. If you fetch a file from an FTAM partner without specifying a file type, the file type is determined by the file attributes in the FTAM partner. For more detailed information about file types when dealing with FTAM partners, see the section “Mapping FTAM attributes to the real file system” on page 114”.

-t (default value with openFT partners)
The file contains text with variable-length records. Records end with the linefeed character \n.

Maximum record length = 32767 bytes

-u The file contains user-structured binary data with variable-length records. Each record starts with 2 bytes which contain the length data of the record.

Maximum record length = 32767 bytes

-b The file contains user-structured binary data with variable-length records.
If you specify the *-b* switch together with *-r* (maximum record length), the file contains binary data with record length specified for *-r*. The size of the send file must be a multiple of this record length.

-x The send file is transferred in a transparent file format and is stored in the destination system, i.e. this is a file whose attributes are transparent for the local system. The local system here acts as a storage and/or transport medium.

If a file is transparently retrieved with *-x* for local buffering, then it must be sent again to the remote system in binary form (i.e. with *-b*).

-o | **-e** | **-n**
Indicates whether the destination file is to be newly created, overwritten, or extended.

-o (default value)
The destination file will be overwritten, or newly created if it does not already exist.

-e The transferred file will be appended to an existing destination file. If this destination file does not exist, it will be newly created.

-n The destination file will be newly created and written. If the destination file already exists, the request will be rejected. In this way, you can protect a file from being overwritten inadvertently.

-k Indicates that identical characters repeated consecutively are to be transferred in compressed form (byte compression). In the case of connections to partners which do not support this type of compression, no compression are used automatically.

-z Indicates that zip compression is used. In the case of connections to partners which do not support this type of compression, byte compression (corresponds to the option *-k*) or no compression are used automatically.

-c Indicates that the data are also encrypted for file transfer. The encryption of the request description data (see page 57) is not affected by this option. If the partner system does not support data encryption, the request is rejected.

-N Suppresses result messages being deposited in the mailbox of the user who issued the request. *-N* is the same as *-m=n*, but is still supported for compatibility reasons.

-S Suppresses file transfer messages to *stderr*.

-m=n | **-m=f** | **-m=a**
This indicates whether the result message is to be deposited in the mail box of the user who issued the request.
With some systems, the mail cannot be delivered if the login name is longer that 8 bytes.

n (default value) The result message is not deposited in the mailbox (identical to the *-N* option).

f The result message is only deposited in the mailbox in the event of errors.

a The result message is always deposited in the mailbox.

file name partner**!**[file name] |
partner**!**[file name] file name
specifies the source and destination. The syntax depends on the direction of transfer selected and if pre- or postprocessing commands are used.

Sending without pre-/postprocessing

Source	**Destination**
local file name	partner**!**[*remote* file name]

Fetching without pre-/postprocessing

Source	Destination
partner**!**[*remote* file name]	*local* file name

Sending and fetching with pre- or postprocessing

If you want to perform pre- or postprocessing, then you must enter an operating system command instead of the local or remote file name (in the syntax of the corresponding system):

Sending with preprocessing

Source	Destination
"**|***local* command"	partner**!**[*remote* file name]

Sending with post-processing

Source	Destination
local file name	Partner**!"|***remote* command"

Fetching with preprocessing

Source	Destination
Partner**!"|***remote* command"	*local* file name

Fetching with post-processing

Source	Destination
Partner**!**[*remote* file name]	"**|***local* command"

You can also combine preprocessing and postprocessing in the same request.

A maximum of 712 characters may be specified both for *source* and *destination* (maximum 512 characters for the file name and maximum 200 for the partner). Please note that the maximum lengths of file names are system-dependent; for example, in Unix systems it is 512 and in Windows systems a maximum of 256 characters.

local file name

Sending: Name of the local file. The file name may include an absolute or relative path name.

Fetching: Name of the receiving local file. The file name may include an absolute or relative path name.

However, the *ft* command will not create a directory which does not already exist.
If the file name ends with %unique or %UNIQUE, this string is replaced by a string which changes for each new call.
In addition, a suffix separated by a dot may be specified after %unique or %UNIQUE, e.g. `file1%unique.txt`.

partner

partner is the name of the partner system in the partner list or the address of the partner system. For details concerning address specification, see section “Defining the partner computer” on page 90.

remote file name

remote file name can be either absolute or relative to the remote login admission. If the file name in the remote system has been predefined in an FT profile, it must not be specified here. If the file name contains blanks, they must be enclosed in double quotes (e.g. "file name").
If the file name ends with %unique or %UNIQUE, this string is replaced by a string which changes for each new call. In addition, a suffix separated by a dot may be specified after %unique or %UNIQUE if the partner is a Unix or Windows system.

If the partner system is running openFT for BS2000/OSD, elements from PLAM libraries may also be specified here (Syntax: Libname/Element type/Element name).

Icommand for *file name*

command is any command on the local or remote system. The "|" character (vertical bar) must always be placed before the command. The "|" character must always be escaped by either a backslash (\) or double quotes ("), i.e. "Icommand" should always be enclosed in double quotes.

In the case of preprocessing, openFT transfers the data output at the standard output by the command as a file. You can also output the data created by preprocessing in a temporary file created by openFT.

During postprocessing, you can have the transferred data stored in a temporary file created by openFT.

You can find out the name of this temporary file and pass it to preprocessing or postprocessing with the variable %TEMPFILE. See the section “Preprocessing and postprocessing” on page 100.

If command execution takes longer than ten minutes, a timeout occurs on partners using versions of openFT prior to V8.1 and command execution is regarded as having failed. This restriction no longer applies to partners using openFT V8.1 or later.

Remote command execution in Unix and Windows systems starts in the user's $HOME directory or home directory respectively.

The PATH variable is used as follows in the search path for preprocessing and postprocessing commands in Unix systems:

– Default instance:
  :/opt/openFT/bin:/bin:/usr/bin:/opt/bin
– Other instance:
  :/var/openFT/instance/openFT/bin:/bin:/usr/bin:/opt/bin
  where *instance* is the name of the relevant instance.

This means that the system first searches in the current directory (first ":"). Before calling a "real" preprocessing or postprocessing command you can switch to another directory as follows:

cd *path-name*;*command*

path-name is then used as the current directory. There must not be a blank between the semicolon and the command.

If the string "|&" comes before the preprocessing/postprocessing command instead of the character "|", the openFT request is restartable (see section “Preprocessing” on page 43 and section “Postprocessing” on page 44).

transfer admission | **@d** | **@n** |
user ID[**,**[account][**,**password]]]

To be able to send a file to a remote system or to fetch one from it, you must furnish the remote system with proof of identity. For this purpose, you will need login admission in the syntax valid for the remote system. You can specify transfer admission

– as an FTAC transfer admission if FTAC is used in the remote system,

– or as a login/LOGON admission in the syntax used by the remote system (user ID, possibly together with account or password).

For details, see section “Transfer admission” on page 94.

@d for *transfer admission*

Specifying *@d* (blanked transfer admission) causes openFT to query the transfer admission on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the transfer admission.

@n for *transfer admission*

By entering *@n*, you specify that the remote system requires no login admission.

A binary password and a binary transfer admission must be entered in hexadecimal form x'...' or X'...'. If you enter the password directly, remember to insert a backslash (\) to escape the single quotes if you did not enclose the remote login admission in double quotes, for example: X\'c6d9e4c5\'.

password not specified

Omitting the password necessary for admission causes openFT to query the password on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the password. In this case, single quotes must not be escaped by a backslash (\).

Nevertheless, you have to specify the commas, e.g.:

```
ft file partner!file user-id,,
```

or

```
ft file partner!file user-id,account,
```

neither *transfer admission* nor *user ID* specified
: causes the same as *@d*, i.e. openFT queries the transfer admission on the screen after the command is entered. Your (blanked) entry is always interpreted as transfer admission and not as user ID.

-p=[password]
: If the file in the remote system is protected by a write password, you must enter this password when sending a file. If the file is protected by a read password, then this password must be specified when fetching a file from the remote system.

  A binary password must be entered in hexadecimal form x\'...\' or X\'...\'. This is of relevance for links to openFT for BS2000/OSD, because BS2000 supports the definition of hexadecimal passwords. If you enter the password directly, remember to insert a backslash (\) to escape the single quotes if you did not enclose the remote login admission in double quotes, for example: X\'c6d9e4c5\'.

  password not specified
  : Specifying *-p=* causes openFT to query the write or read password on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the password. In this case, single quotes must not be escaped by a backslash (\).

-di is specified, if the data integrity of the transferred file is to be checked by cryptographic means. With it, harmful data manipulations on the transmission network are identified. In case of an error openFT performs an error recovery for asynchronous transfer requests.

If the partner system does not support the check of data integrity (e.g. openFT < V8.1), the request is denied.

For requests with data encryption (option *-c*), data integrity is automatically checked. Testing mechanisms of the transfer protocols in use automatically identify transfer errors in the network. For this purpose you do not have to specify the *-di* option.

-lc=CCS name
: (local coding) specifies the type of coding (character set) to be used to read or write the local file. *CCS name* must be known in the local system.

  The default value is the character set defined by the FT administrator.

  Details about the CCS name and the associated code tables can be found in section “Code tables and coded character sets (CCS)” on page 85.

-rc=CCS name
(remote coding) specifies the type of coding to be used to read or write the remote file. *CCS name* must be known in the remote system.

The default value is the character set defined in the remote system via XHCS (BS2000/OSD) or the openFT operating parameters (other platforms).

The option *-rc* is supported only by the openFT protocol and partners with openFT V10.0 or higher. Please note that not all partner systems support all the character sets that are possible in the local system.

Details about the CCS name and the associated code tables can be found in section “Code tables and coded character sets (CCS)” on page 85.

-ls=’follow-up processing’
Here you can specify a command which will be executed in the local system following a **successful transfer** operation.
Further information is given in the section “Commands for follow-up processing” below.

-lf=’follow-up processing’
Here you can specify a command which will be executed in the local system if a transfer operation is **terminated** as a result of an **error**.
Further information is given in the section “Commands for follow-up processing” below.

-rs=’follow-up processing’
Here you can specify a command in the syntax of the remote system. Following a **successful transfer** operation, this command is executed in the remote system under the specified login.
Further information is given in the section “Commands for follow-up processing” below.

-rf=’follow-up processing’
Here you can specify a command in the syntax of the remote system. This command will be executed in the remote system under the specified login if a transfer operation that has already started is **terminated** as a result of an **error**.
Further information is given in the section “Commands for follow-up processing” below.

Commands for follow-up processing

- The total number of entries for local follow-up processing, i.e. for *ls* and *lf*, may not exceed 1000 characters.

  The total number of characters for remote follow-up processing, i.e. for *rs* and *rf*, may not exceed 1000 characters, but this maximum value may be lower if a FT version < V10 is used in the remote system.

- The entries for follow-up processing must be enclosed in single quotes (') or double quotes (").
  If the entry for follow-up processing also contains single quotes (’), it is recommended to enclose the entire entry in double quotes (“).
  The single quotes in the follow-up processing command (e.g. single quotes in a BS2000 password) can then be written as expected in the partner system (e.g. BS2000).

- When starting follow-up processing in the local or remote system, the specified variables are first substituted, and the follow-up processing commands are then executed. The following variables are permitted:

  %FILENAME
  File name in the relevant system. The entry is automatically taken from the command. If you specified the variable %UNIQUE (or %unique) for the remote file name during transfer, the %FILENAME variable contains the already converted (i.e. unique) file name.

  %PARTNER
  Name or address of the partner system in long form, i.e. with dynamic partners, all address components are taken (protocol prefix, port number, selectors, ...). The behavior is different for local and remote follow-up processing. For local follow-up processing, the partner name specified in the call is used. For follow-up processing in the remote system, %PARTNER is substituted by the name of the initiator system (with the name as known in the partner system).

  %PARTNERAT
  Name or address of the partner system in short form, i.e. with dynamic partners, only the *host* address component is taken, see page 143. In addition, each character is replaced by a ’@’ if it is neither a letter nor a digit or a period.

%RESULT

is replaced by the message number applicable to the request, as required by the system concerned.
If, for example, a send request is successfully executed, the value of %RESULT in the local system is the message number 0 (for openFT as of V10).

If the partner system is an openFT for BS2000/OSD system, you may also use the variables %ELEMNAME, %ELEMVERS and %ELEMTYP.

- Special considerations with Windows systems
  - In the case of follow-up processing on a remote Windows system, only the system environment variables are available, not the user variables. In addition, the user-specific Registry entries are not loaded before follow-up processing is executed.
  - Any program can be started as follow-up processing in Windows, e.g. a shell command, a program (*.exe* or *.com*) or a batch procedure (*.bat* or *.cmd*). If the command requires a path specification, then use the absolute path.
  - Before calling the follow-up processing in a remote Windows system, it is also possible to switch to another directory as follows:

    `cd` *path-name*`;`*command*

    path-name is then used as the current directory. There must not be a blank between the semicolon and the command. *path-name* must not be a directory which is addressed using a UNC name.
  - If you wish to execute shell-internal Windows commands (e.g. *move* or *copy*), remember that you must specify the command processor *cmd.exe /c* at the start of the command.
- Follow-up processing in the local system and follow-up processing in a remote Unix system does **not** involve execution of the sequence of commands stored in the *.profile* file. Only the default values of the $HOME, $LOGNAME, $PATH, and $USER shell variables are available, as well as the values of the $LANG and $TZ variables set by *root*.
- The search path (PATH variable) for follow-up processing commands is preceded by the component */var/openFT/instance/openFT/bin*, where *instance* means the name of the corresponding instance.

- With requests for FTAM or FTP partners, the follow-up processing function is not available in the remote system (exception: -rs=´∗DELETE´ for FTAM receive requests to delete the send file after successful processing). If FTAC is used in the remote system, this restriction can be avoided by creating an FT profile in the remote system and defining follow-up processing for it.
- When specifying BS2000 commands, remember to insert a slash (/) at the beginning of the command.

-r=v[record length] | **-r=f**[record length] | -**r=u**[record length] | -**r=**record length
Specifies the record format and the record length. This also enables records that are longer than the default value to be transferred. However, you must bear in mind that not every record length can be processed in all partner systems.

If you have selected the file type *b* (binary), *record length* it the value for all records of the send file.

Maximum value for *record length*: 32767 bytes.

With FTAM partners, the maximum record length specification is not valid unless the file type is set explicitly to *t*, *b* or *u*.

It is also possible to output the record format, see also page 227:

v variable record length, *record length* defines the maximum value

f fixed record length, *record length* then applies to all records

u undefined record length

The combinations `-u -r=f`*recordlength* and `-u -r=u`*recordlength* are not permitted.

If *-r* is omitted then the following default values apply for the record format:

Option	Default value	Corresponds to
-b	u (undefined)	-r=u...
-t	v (variable)	-r=v...
-u	v (variable)	-r=v...

-tff=b | **-tff=s**
Specifies the format of the destination file.

b The destination file is to be saved as a block-structured file. This means, for example, that a file can be transferred to BS2000 and stored there as a PAM file. If you specify *-tff=b*, you must also specify the option *-b* (binary).

s The destination file is to be saved as a sequential file and the record format is to be retained. This allows an ISAM file or PAM file to be fetched from BS2000, for instance.

-tff must not be specified at the same time as *-trf*.

-trf=u Specifies that the file is to be transferred as a sequential file and that the record format of the destination file is to be undefined, i.e. the record structure of the send file is lost. If the file is being transferred to a BS2000 or z/OS system, one block is written per transfer unit.

-trf must not be specified at the same time as *-tff*.

neither *-tff* nor *-trf* specified
The destination file is to be stored in the same format as the send file.

-tb=n | **-tb=f** | **-tb=a**
Activates/deactivates tabulator expansion and the conversion of blank lines into lines with one character for non-FTAM partners for a single output send request.

The following parameters are provided:

n (on)
Tabulator expansion and blank line conversion are activated.

f (off)
Tabulator expansion and blank line conversion are deactivated.

a (automatic, default value)
Tabulator expansion and blank line conversion are activated if a file is sent to a BS2000, OS/390, or z/OS system.

No tabulator expansion or blank line conversion is performed for outbound receive requests.

If *ft* is used as a preprocessing command, then tabulator expansion and blank line conversion are always deactivated.

The following parameters *-av, -ac*, *-am*, *-lq* and *-cp* are provided exclusively for communication with FTAM partners. openFT thus supports the parameters defined in the FTAM standard. These parameters enable you to define the attributes of the destination file while issuing a file transfer request.

These parameters are ignored for requests involving openFT and FTP partners, but the file transfer is still carried out.

-av=i | **-av=d**
: Indicates the availability of the destination file. This parameter can have one of two values: *immediate* or *deferred*. A file may be *deferred* if it has been archived, for example. The partner is responsible for interpreting the term *deferred*. The FTAM partner conventions must therefore be observed here.

  The following values are possible:

  i The destination file attribute is set to *immediate*.

  d The destination file attribute is set to *deferred*.

  av is not available for requests involving FTAM partners that do not support the storage group. In this case, the request is executed, but the entry for *av* is ignored.

-av not specified
: The availability file attribute is set to a system-specific default value. In Unix systems, this is the value *immediate*.

-ac=new account
: With FTAM partners, this indicates the number of the account to which file storage fees are to be charged. This parameter must be set in accordance with partner system conventions.

  ac is not available for requests involving FTAM partners that do not support the storage group. In this case, the request is executed, but the entry for *ac* is ignored.

-am=[**r**][**i**][**p**][**x**][**e**][**a**][**c**][**d**] | **-am=@rw** | **-am=@ro**
: This sets the access rights of the destination file, provided the security group is available.

  The following values can be specified for access mode:

  r, *i*, *p*, *x*, *e*, *a*, *c*, *d*, any combination of these values, *@rw*, or *@ro*.

  r means that the file can be read.

r not specified
: The file cannot be read.

i means that data units, such as records, can be inserted in the file.

i not specified
: No data units can be inserted in the file.

p means that the file can be overwritten.

p not specified
: The file cannot be overwritten.

x means that data can be appended to the file.

x not specified
: The file cannot be extended.

e means that data units, such as records, can be deleted from the file.

e not specified
: No data units can be deleted from the file.

a means that the file attributes can be read.

a not specified
: The file attributes cannot be read.

c means that the file attributes can be changed.

c not specified
: The file attributes cannot be changed.

d means that the file can be deleted.

d not specified
: The file cannot be deleted.

@rw is the short form of the common access rights *read-write* (*rpxeacd*), and thus simplifies input.

@ro is the short form for the common access rights *read-only* (*rac*), and thus simplifies input.

If the partner system is a Windows system, you cannot change the access rights of the destination file.

In Unix systems or in BS2000, the following access rights can be set for a file:

Access mode	Short form	Unix system	BS2000	Access rights
rpxeacd	@rw	rw*	ACCESS=WRITE	read-write
rac	@ro	r-*	ACCESS=READ	read-only
pxeacd		-w*	Only with BASIC-ACL (Access Control List)	write-only
ac		--*	Only with BASIC-ACL (Access Control List)	none

* The x bit is not changed by *ft*.

-am is not available for requests involving FTAM partners that do not support the security group. In this case, the request is executed, but the entry for *-am* is ignored.

-am not specified
: The default values of the FTAM partner system apply.

-lq=legal qualification
: This specifies a legal qualification for the destination file (similar to a copyright). This may not exceed 80 characters.

-lq is not available for requests involving FTAM partners that do not support the security group. The request is executed, but the entry for *-lq* is ignored.

-cp=[password]
: If a password is required in order to create a file on a remote system, this password must be specified here. It can be up to 64 characters long. A binary password must be specified in hexadecimal format in the form x\'...\' or X\'...\'. If you do not specify a file creation password, but you do enter a file access password for *-p=password*, the file creation password is identical to the file access password. The file creation password is of no significance when retrieving a file.

password not specified
: Specifying *-cp=* causes openFT to query the file creation password on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the password.

-**pr=n** | **-pr=l**
indicates the priority of the request:

n (normal)
the request has the priority "normal" (default value).

l (low)
the request has the priority "low".

-**sd=**start date
indicates the earliest date at which the file transfer is to be started. Possible values:

yyyymmdd
e.g. 20091031 for the start transfer on October 31, 2009. The largest possible value for the date is 20380119 (January 19, 2038).

+ddddd
e.g. +2 for start of transfer 2 days after issuing the request. You can delay file transfer by 999 days at the most. You can specify at most five figures for the delayed date. The value is limited by the number of days up to 19.01.2038.

-**st=**start time
specifies the earliest time at which file transfer is to be started (due to the nature of the system, the start time may deviate 5 minutes from the specified time). Possible values:

hhmm
e.g. 1430 for start of transfer at 14.30 hrs.

+hhmm
e.g. +0230 for start of transfer 2 hours and 30 minutes after issue of the request. The maximum delay you may specify is 99 hours and 59 minutes.

The start time must not be specified as relative if the start date has been specified as absolute. For a relative start date and start time, the start time is calculated from the total of the two entries, i.e. if a request is issued at 10.07. at 15.00 hrs. with *-sd=+1* and *-st=+1000*, the request is not started until 12.07. at 01.00 hrs.

If you enter a start date without a start time, transfer is started at 0.00 hrs. on the date specified. If you enter a start time without a start date, the time applies to the current date. If you specify a request with *-st=1000* at 15.00 hrs then the request is run immediately.

-**cd=**cancel date

Specifies the date on which the request is to be deleted. If the request is active at the time specified, it is aborted. Possible values:

yyyymmdd

e.g. 20091231 for cancellation of the request on December 31, 2009. The specified time must not lie in the past. The largest possible value for the date is 20380119 (January 19, 2038).

+ddddd

e.g. +2 for cancellation of the request 2 days after its issue. The maximum delay you may specify is 999 days. You can specify at most five figures for the delayed date. The value is limited by the number of days up to 19.01.2038.

-**ct=**cancel time

Specifies the time at which the request is to be deleted (due to the nature of the system, the start time may deviate 5 minutes from the specified time). The specified time must not lie in the past. If the request is active at the time specified, it is aborted. Possible values:

hhmm

e.g. 1430 for cancellation of the request at 14.30 hrs. The specified time must not lie in the past.

+hhmm

e.g. +0230 for cancellation of the request 2 hours and 30 minutes after its issue. The maximum delay you may specify is 99 hours and 59 minutes.

If you enter a cancel date without a cancel time, the file transfer is canceled at 23.59 hrs on the date specified. If you specify a cancel time without a cancel date, the time applies to the current date.

The cancel time must not be specified as relative if the cancel date has been specified as absolute. For a relative delete date and delete time, the delete time is calculated from the total of the two entries, i.e. if a request is issued at 10.07. at 15.00 hrs. with -cd=+1 and -ct=+1000, the request is not deleted until 12.07. at 01.00 hrs.

Requests also have a limited lifetime, even if no values are specified for *-cd* and *-ct*. This lifetime is set by the FT administrator. You may query the value using the command *ftshwo*. The entry stands for MAX-RQ-LIFE. Specifying *-cd* and *-ct* disables the MAX-RQ-LIFE entry.

Examples

1. The text file *doc.one* is sent by user *jack* to the BS2000 computer with the symbolic name *bs2r1*. Here, it is stored under the login name *jim* with account number *a1234ft* and password C'*pwd*'. The file should then be printed.

```
ft␣doc.one␣bs2r1!doc.one␣jim,a1234ft,C\'pwd\'\
␣-rs="/PRINT-FILE␣%FILENAME,LAYOUT-CONTROL=PARAMETERS\
(,CONTROL-CHARACTERS=EBCDIC)"
```

2. A file is to be fetched from BS2000, where openFT-AC for BS2000/OSD is running, to Unix system. The file name has been predefined in an FT profile, which can be accessed with the access authorization '*fortheRM6*'. In the Unix system, the file is to be stored under the name *test/track.f* as a type *u* file (user format).

```
ft␣-u␣bs2!␣test/track.f␣'fortheRM6'
```

3. The file *source.lst* is sent to the BS2000 computer *bs2r1*. Here, it is stored under the login name *jim* with account number *a1234ft* and password *C'pwd'* under the file name *lst*. Then, as follow-up processing, the file is to be printed out in BS2000 and then deleted. The source file in the local system is likewise deleted.

```
ft␣-source.lst␣bs2r1!lst␣jim,a1234ft,C\'pwd\'\
␣-ls='rm␣source.lst'\
␣-rs='/PRINT␣lst,DELETE-FILE=YES'
```

4. The text file *letter* is sent to the login name *jim* with the password *jimspass* in the FTAM partner with the symbolic name *ftampart*.

```
ft␣letter␣ftampart!letter␣jim,,jimspass
```

5. The text file *locfile* is to be sent to the Unix computer *ux1*. Here, it is to be stored under the login name *charles* with the password *secret* under the file name *remfile*. Then, as follow-up processing, the file is to be printed out if transferred successfully; if not, the *prog* program is to be started in the remote system. As parameters, the program receives the name of the source file and the message number. The parameters are specified using variables. If the request is not completed after 5 hours, it is deleted from the request queue.

```
ft␣locfile␣ux1!remfile␣charles,,secret -r=100\
␣-rs='lpr remfile' \
␣-rf='prog %FILENAME %RESULT' \
␣-ct=+0500
```

If file transfer is not successful, e.g. because the record length was greater than 100 bytes, follow-up processing is executed as follows:

```
prog remfile 2210
```

6. The file *locfile* is sent to the z/OS partner *zospart*. Here, the script PT (e.g. with a print job) is to be executed as follow-up processing under the user ID OPUSER.

```
ft locfile zospart!remfile OPUSER,account,password\
 -rs="alloc dsname('OPUSER.PT')"
```

7. Example of specifying domain user IDs in a remote Windows system:

```
ft file2 Win01!file2 mydomain\\miller,,secret
```

8. This example shows the use of restartable pre- and postprocessing commands. The local directory *dir*, along with all its files, is to be transferred to a remote Unix computer using the symbolic name *ftunix*. The current version of openFT should also be running on the remote computer. After the transfer, dir should be available on the remote system under the ID to which the access admission *copydir1* belongs. The directory *dir* must be located on the local computer in *$HOME*. Please note that no file name prefix is allowed to be defined in the profile. Details on *ft_tar* are located in the appendix (page 420).

```
ft "|&ft_tar -cf -  dir" ftunix!"|&ft_tar -xf - " copydir1 -b
```

5.5 ftcanr - Cancel asynchronous requests

You can use the *ftcanr* command to cancel asynchronous requests which are in the course of being processed or which are waiting to be processed in the request queue. As an ordinary FT user, you can only cancel requests entered under your own login name.

If file transfer requests have already been started, the status of the destination file may be undefined.

Format

ftcanr -h |
[-f]
[-ua=<user ID 1..32> | @a]
[-ini=l | -ini=r | -ini=lr | -ini=rl]
[-pn=<partner 1..200>]
[-fn=<file name 1..512>]
<request ID 1..2147483647> [<request ID 1..2147483647> ...] | @a

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-f You can only call this option as FT administrator.

-ua=user ID | **@a**
You use *-ua* to indicates the user ID for which requests are to be cancelled.

user ID
The user can only specify his/her own login name.

@a This option is only significant for the FT administrator.

-ua= not specified
Your login name is used as the selection criterion.

-ini=l | **-ini=r** | **-ini=lr** | **-ini=rl**

You use *-ini* to indicate the initiator for which you want to cancel requests. You can specify *l*, *r*, *lr*, *rl*:

l Only requests initiated locally are cancelled.

r Only requests initiated remotely are cancelled.

lr, rl Both local and remote requests are cancelled.

-ini not specified

The initiator is not used as a selection criterion (corresponds to *lr* or *rl*).

-pn=partner

You use *-pn* to specify the partner system for which you want to cancel requests. *Partner* is the name or address of the partner system. You should specify the partner in the same form as in the request allocation or as in the output from the *ftshwr* command.

-fn=file name

You use *-fn* to specify the name of the file for which requests are to be cancelled. Requests which access this file in the local system are cancelled.

You must specify the file name which was used when the request was issued and which is output for the *ftshwr* command. Wildcards are not permitted in file names.

request ID1 [request ID2] [request ID3] ... | **@a**

For *request ID*, enter the number of the request to be cancelled. Leading zeros may be omitted. The request identification *request ID* may be obtained from the request receipt acknowledgment displayed on the screen, or using the *ftshwr* command if you have forgotten the *request ID*. You can also specify a number of request identifications at the same time. If, in addition to *request ID*, you specify other selection criteria, a request with the specified *request ID* is only cancelled if it also satisfies the other conditions.

@a specified as *request ID*

@a selects all requests.

If request IDs were specified and the other selection criteria specified are not satisfied by the requests, the request is not cancelled and the following new error message is issued:

`ftcanr: Request` *request ID* `not found`

request ID is the identification of the last unsuitable request.

Examples

1. The asynchronous request with request identification 65546 should be deleted.

```
ftcanr␣65546
```

2. All local requests to the partner *ux1* which relate to the file *file1* should be deleted.

```
ftcanr -pn=ux1 -fn=file1 -ini=l @a
```

5.6 ftcredir - Create remote directories

You use *ftcredir* to create a new directory on a remote system. This is only possible if the remote system supports this function.

Format

ftcredir -h |
<partner 1..200>![<file name 1..512>]
[<transfer admission 8..67> | @n | @d |
<user ID 1..67>[,[<account 1..64>][,[<password 1..64>]]]]
[-p=[<management password 1..64>]]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

partner**!**[file name]
Specifies what directory is to be created on what computer.

partner
Partner is the name of the partner system in the partner list or the address of the partner system. For details on address specifications, see section “Defining the partner computer” on page 90.

file name
Name of the directory that is to be created. You can specify the name absolutely or relative to the remote login authorization. If the name in the remote system is predefined by an admission profile then it may not be specified here.

If openFT for BS2000/OSD is running on a partner system then an empty PLAM library is created.

transfer admission | **@n** | **@d** |
user ID[**,**[account][**,**[password]]]
Before you can modify the attributes of a file on a remote system, you must first identify yourself at the system. To do this, you need an authorization in the syntax used at the remote system.

You can specify this transfer admission

– as an FTAC transfer admission if FTAC is used in the remote system,

– or as a login/LOGON admission in the syntax used by the remote system (user ID, possibly together with account or password).

For details, see section “Transfer admission” on page 94.

@n for *transfer admission*
With *@n* you specify that the remote system does not demand a login authorization.

@d for *transfer admission*
If you specify *@d* (blanked) then the transfer admission is queried on the screen after the command has been sent. Your input is invisible to prevent unauthorized persons from seeing the transfer admission.

A binary password or binary transfer admission must be specified in hexadecimal format in the form x'...' or X'...'. If you enter the password directly, remember to invalidate the single quotes with a backslash (\) unless you have enclosed the remote login authorization in double quotes, for example X\'c6d9e4c5\'.

password not specified

If you omit a password which is required for authorization then it is queried on the screen after the command has been sent. Your input is invisible to prevent unauthorized persons from seeing the password. In this case, quotes must not be invalidated with a backslash (\).

Please note that you still have to enter the commas, e.g.:

```
ftcredir partner!file identification,,
```
or
```
ftcredir partner!file identification,account,
```

neither *transfer admission* nor *user ID* specified
This has the same effect as *@d*, i.e. the transfer admission is queried on the screen after the command has been sent. openFT always interprets your (hidden) input as a transfer admission and not as a user ID.

-p=[management password]

If you want to create a new directory in a password-protected PLAM library then you must specify the password here.

The password can also be specified in hexadecimal format in the form x'...' or X'...'. This is of relevance in the case of a connection with openFT for BS2000/OSD since it is possible to define hexadecimal passwords in BS2000. If you enter the password directly, remember to invalidate the single quotes with a backslash (\), for example: -p=X\'c6d9e4c5\'.

management password not specified

If you specify *-p=* then the password is queried on screen after the command has been sent. Your input is invisible to prevent unauthorized persons from seeing the password. In this case, quotes must not be invalidated with a backslash (\) .

Examples

1. In the remote Unix system *ux1*, you want to create the directory *dir1*. The identification in *ux1* is protected via the transfer admission *userremote*.

   ```
   ftcredir ux1!dir1 userremote
   ```

2. In the remote Windows system *win1*,you want to create the directories *dir1\dir2* and *dir2* is to be a subdirectory of *dir1*. Neither of these directories exists yet. The directories are to be created in the existing directory *exdir* under the ID *jerry* with the password *secret*.

   To do this, you enter the following commands:

   ```
   ftcredir win1!exdir/dir1 jerry,,secret
   ftcredir win1!exdir/dir1/dir2 jerry,,secret
   ```

   The first command is necessary because if you only entered the second command (`ftcredir win1!exdir/dir1/dir2 jerry,,secret`) then the directory *dir1* will not yet exist in the remote system and you will see the error message:

   ```
   Remote system: Higher-level directory not found
   ```

3. In the remote BS2000 system *bs2*, you want to create the PLAM library *user.lib*, the ID is *jimbs2* with the account *j123456* and the password *jimpass*.

   ```
   ftcredir bs2!user.lib jimbs2,j123456,jimpass
   ```

5.7 ftcrep - Create an FT profile

ftcrep stands for "create profile". This command can be used by any user to set up FT profiles for his or her login name.

When it is created, the profile is given a timestamp that is updated each time the profile is modified (e.g. using *ftmodp*).

Format

```
ftcrep  -h |
          <profile name 1..8> | @s
          <transfer admission 8..32> | @n
        [ -ua=[<user ID 1..32>] [,[<password 1..20> | @n ]] ]
        [ -v=y | -v=n ] [ -d=yyyymmdd ]
        [ -u=pr | -u=pu ]
        [ -priv=y | -priv=n ]
        [ -iml=y | -iml=n ]
        [ -iis=y | -iis=n ] [ -iir=y | -iir=n ]
        [ -iip=y | -iip=n ] [ -iif=y | -iif=n ]
        [ -ff=[t][m][p][r][a][l] | -ff=c ]
        [ -dir=f | -dir=t | -dir=ft ]
        [ -pn=<partner 1..200>,...,<partner(50) 1..200> | -pn= ]
        [ -fn=<file name 1..512> | -fn= ]
        [ -fnp=<file name prefix 1..511> ]
        [ -ls= | -ls=@n | -ls=<command1 1..1000> ]
        [ -lsp=<command2 1..999> ] [ -lss=<command3 1..999> ]
        [ -lf= | -lf=@n | -lf=<command4 1..1000> ]
        [ -lfp=<command5 1..999> ] [ -lfs=<command6 1..999> ]
        [ -wm=o | -wm=n | -wm=e | -wm=one ]
        [ -c=y | -c=n ]
        [ -txt=<text 1..100> ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

profile name | **@s**
is the name you wish to assign to the FT profile. This name can be used to address the FT profile, for example when it is to be modified or deleted. Be sure not to confuse the profile name with the transfer admission (see below). The profile name must be unique among all the FT profiles under

your login name, or FTAC will reject the *ftcrep* command and issue the message *FT profile already exists*. To have the profile names you have already assigned displayed, you can issue the *ftshwp* command (without options).

@s for *profile name*
: Creates the standard admission profile for the user ID. You must specify *@n* as the transfer admission, because a standard admission profile in a request is addressed using the user ID and password.

  You must not specify the options *-v*, *-d* and *-u* with a standard admission profile.

transfer admission | **@n**
: replaces the login authorization for your Unix system otherwise required in FT requests. When this transfer admission is specified in an FT request, FTAC applies the access rights defined in this FT profile.

  transfer admission
  : The transfer admission must be unique within your Unix system so that there are no conflicts with transfer admissions defined by other FTAC users with other access rights. If the transfer admission you select has already been assigned, FTAC rejects the *ftcrep* command and issues the message:
    `Transfer admission already exists.`

    You can also define a binary admission with any characters, including non-printing characters. To do this, you must specify the transfer admission in hexadecimal format in the following form: x\'...\' or X\'...\', e.g. x\'f1f2f3f4f5f6f6f8\'.

  @n for *transfer admission*
  : By entering *@n*, you create an FT profile without a transfer admission.

    If the profile is not a standard admission profile, it is locked until you assign a valid transfer admission with *ftmodp*.

    You must specify *@n* when you create a standard admission profile.

transfer admission not specified

If you do not specify the transfer admission in the command, FTAC prompts you to enter the transfer admission after the command has been sent. Your entry is not displayed to prevent unauthorized persons from seeing the transfer admission. To exclude the possibility of typing errors, the program expects you to enter the transfer admission a second time as an entry check.

-ua=[user ID][**,**[password | **@n**]]

user ID

The user without administrator privileges can specify only his own user ID.

,password

Specifies the password of the login name. A binary password must be specified in hexadecimal format in the form x\'...\' or X\'...\'. The FT profile for the login name is only valid while the password is valid for the login name. If the password is changed, the profile can no longer be used.

This entry may only be specified by the FTAC administrator.

comma only (,) no *password*

Entering comma (,) without *password* causes FTAC to query the password on the screen after the command is entered. The entry is not displayed to prevent unauthorized persons from seeing the transfer admission. In this case, quotes must not be escaped with a backslash (\).

user ID only (without comma and no *password*) specified

the profile is valid for all the passwords for *user ID*.

-ua=␣ specified or *-ua* not specified

the FT profile is created for the individual login name.

-v=y | **-v=n**

defines the status of the transfer admission.

Possible values are:

y (default value)

the transfer admission is not disabled (it is valid).

n the transfer admission is disabled (it not valid).

-v must not be specified with a standard admission profile.

-d=yyyymmdd
: specifies the period during which the transfer admission can be used. The FT profile is disabled when this period has expired.
You can specify an eight-digit date (e.g. 20170602 for June 2, 2017). The transfer admission can no longer be used after 0:00 hours on the specified day. The largest possible value which can be specified as the date is 20380119 (January 19, 2038).

  -d must not be specified with a standard admission profile.

-d not specified (default value)
: no period is specified for using the transfer admission.

-u=pr | **-u=pu**
: with *-u*, you can control how FTAC reacts when someone attempts to create an FT profile with the same transfer admission. Normally, the transfer admission must be disabled immediately.

  Transfer admissions that do not require as much protection are designated as public. This means that they are not disabled, even if a user attempts to assign another transfer admission of the same name.

  pr (default value)
  : the transfer admission is disabled as soon as someone under another login name attempts to specify a transfer admission of the same name (private). In this case, the values for *-u* and *-d* are set to their default values at the same time.

  pu the transfer admission is not disabled, even if someone attempts to specify a transfer admission of the same name (public).

  -u must not be specified with a standard admission profile.

-u not specified
: The previous setting remains unchainched.

-priv=n | **-priv=y**
: As a user, you can only revoke an existing privileged status, *y* is not permitted.

  n (default value)
  : The FT profile is not privileged (initially).

  y For the FTAC administrator only: The FT profile is privileged.

-iml=y | **-iml=n**

-iml (ignore max. level) is used to specify whether the FT profile is to be restricted by the values in the admission set. You can override your own the entries (the MAX. USER LEVELS) for requests using this FT profile.

If the FT profile is also privileged by the FTAC administrator, the values of the FTAC administrator (the MAX. ADM LEVELS) can also be ignored. This FT profile would then allow *inbound* basic functions which are disabled in the admission set to be used. Possible values are:

y allows the values in the admission set to be ignored.

n (default value)
restricts the functionality of the profile to the values in the admission set.

-iis=y | **-iis=n**

-iis (ignore inbound send) allows the value for the basic function *inbound send* in the admission set to be ignored (for details, see *-iml*).

y allows the basic function *inbound send* to be used even if it is disabled in the admission set. At the same time, the component "display file attributes" of the basic function *inbound file management* can also be used.

Specifying this option is enough as long as the basic function *inbound send* was disabled by the user, but if it was disabled by the FTAC administrator, it is also necessary that he/she grant privileged status to the FT profile.

n (default value)
restricts the profile to the value in the admission set for the basic function *inbound send*.

-iir=y | **-iir=n**

-iir (ignore inbound receive) allows the value for the basic function *inbound receive* in the admission set to be ignored (for details, see *-iml*).

y allows the basic function *inbound receive* to be used even if it is disabled in the admission set. At the same time, components of the basic function *inbound file management* can also be used (see table at *-iif*).

Specifying this option is enough as long as the basic function *inbound receive* was disabled by the user, but if it was disabled by the FTAC administrator, it is also necessary that he/she grant privileged status to the FT profile.

n (default value)
restricts the profile to the value in the admission set for the basic function *inbound receive*.

-iip=y | **-iip=n**
-iip (ignore inbound processing) allows the value for the basic function *inbound follow-up processing + preprocessing + postprocessing* in the admission set to be ignored (for details, see *-iml*).

y allows the basic function *inbound follow-up processing + preprocessing + postprocessing* to be used even if it is disabled in the admission set.
Specifying this option is enough as long as the basic function *inbound receive* was disabled by the user, but if it was disabled by the FTAC administrator, it is also necessary that he/she grant privileged status to the FT profile.

n (default value)
restricts the profile to the value in the admission set for the basic function *inbound follow-up processing + preprocessing + postprocessing*.

-iif=y | **-iif=n**
-iif (ignore inbound file management) allows the values for the basic function *inbound file management* in the admission set to be ignored (for details see *-iml*).

y allows the basic function *inbound file management* to be used even if it is disabled in the admission set. Specifying this option is enough as long as the basic function *inbound file management* was disabled by the user, but if it was disabled by the FTAC administrator, it is also necessary that he/she grant privileged status to the FT profile.

n (default value)
restricts the profile to the value in the admission set for the basic function *inbound file management*.

The following table shows which subcomponents of the file management can be used under which conditions.

Inbound file management function	**Values of the admission set or extension in profile**
Display file attributes	Inbound Send (IBS) enabled
Modify file attributes	Inbound Receive (IBR) **and** Inbound File Management (IBF) enabled
Rename files	Inbound Receive (IBR) **and** Inbound File Management (IBF) enabled
Delete files	Inbound Receive (IBR) enabled **and** Write mode = overwrite in profile
Display directories	Inbound File Management (IBF) enabled
Create, rename and delete directories	Inbound File Management (IBF) enabled **and** direction= from partner in profile

-ff=[t][m][p][r][a][l] | -ff=c

-ff defines the FT function for which the FT profile can be used. With the exception of *c*, these letters can be combined in any way (*tm, mt, mr, ...*). *c* must not be combined with other values.

t (transfer) The FT profile can be used for the file transfer functions "Transfer files", "Display file attributes", and "Delete files".

m (modify file attributes) The FT profile can be used for the file transfer functions "Display file attributes" and "Modify file attributes".

p (processing) The FT profile can be used for the file transfer functions "File Preprocessing" or "File Postprocessing". The FT function "Transfer files" must also be permitted.

Specification of *p* has no significance for profiles with a file name prefix (*-fnp=*) or a file name (*-fn=*) since, in this case, the first character of the file name or file name prefix decides whether the profile can only be used for preprocessing and postprocessing ("|") or only for file transfer/file management (no "|").
The use of follow-up processing is not controlled by *-ff=*, but by *-lf=* and *-ls=*.

r (read directory) The FT profile can be used for the file transfer functions "Display directories" and "Display file attributes".

a (administration)
The admission profile is allowed to be used for the "remote administration" function.
-ff=a may only be specified by the FT administrator or FTAC administrator.

l (logging)
The admission profile is allowed to be used for the "ADM traps" function.
-ff=l may only be specified by the FT administrator.

c (client access)
The admission profile is allowed to be used for the "access to remote administration server" function (ADM profile).

The value *c* must not be combined with any other value. *-ff=c* may only be specified by the ADM administrator.

-ff not specified
Corresponds to the specification *-ff=tmr*, i.e. the admission profile can be used for all file transfer functions other than "file processing", but cannot be used for remote administration functions (*a*, *c*) and ADM traps (*l*).

-dir=f | **-dir=t** | **-dir=ft**
specifies for which transfer direction(s) the FT profile may be used.

f allows data transfer only from a remote system to the local system.

t allows data transfer only from a local to a remote system. Directories cannot be created, renamed nor deleted.

ft, tf both transfer directions are allowed.

-dir not specified
transfer direction is not restricted in the FT profile.

-pn=partner[**,**partner2**,** ...] | **-pn=**
You use *-pn* to specify that this admission profile is to be used only for FT requests which are processed by a certain partner system. You can specify the name of the partner system in the partner list or the address of the partner system. For details on address specifications, see section “Defining the partner computer” on page 90.

You can specify more than one partner system (maximum 50) with a maximum total of 1000 characters.

-pn not specified (or **-pn=**)
: means that any remote system can use the FT profile.

-fn=file name | **-fn=**
: *-fn* specifies which file under your login name may be accessed using this FT profile. If you specify a fully qualified file name, only the file with this name can be transferred.
If the file name ends with %unique or %UNIQUE, this string is replaced during the file transfer by a string which changes for each new call. In Unix systems, this string is 14 characters long. In addition, a suffix separated by a dot may be specified after %unique or %UNIQUE, e.g. `file1%unique.txt`. Only the already converted file name is displayed in both the log and the messages.
If *file name* starts with a "|" (pipe character) then it is interpreted as a preprocessing or postprocessing command, see also section “Preprocessing and postprocessing” on page 100.

-fn not specified (or **-fn=**)
: omitting *-fn* means that the FT profile allows unrestricted access to all files under the login name (exception see *-fnp*).

-fnp=file name prefix
: restricts access to a set of files whose names begin with the same prefix. FTAC adds the character string specified as *file-name-prefix* to the file name in the request and attempts to transfer the file with the expanded name. For example, if this option is specified as *-fnp=scrooge/* and the request contains the file name *stock*, the file transferred is *scrooge/stock*).

  In this way, you can designate the files you have released for transfer. If the *-fnp* option was used to specify a prefix, the file name specified in the request must not contain the character string *../*. This disables (unintentionally) changing directories. You should also ensure that there is no chance for a symbolic link to cause a jump to another place in the file tree.

  %unique or %UNIQUE cannot be used for a file name prefix. In the case of a file transfer request, the user can use a file name ending with %UNIQUE (or %UNIQUE.*suffix* or %unique or %unique.*suffix*) to generate a unique file name with the prefix specified here.

  A file name prefix which starts with the | (pipe) character indicates that the FTAC profile can only be used for file transfer with preprocessing and postprocessing, since the file name created using the prefix and the name specified for the *ncopy* or *ft* command also starts with the | character. In this case, no follow-up commands may be specified.

On Unix systems, the shell metacharacters | ; & < > and "newline" may only be specified if they are enclosed in '...' (single quotes) or "..." (double quotes) or if each of them is escaped with "\" (backslash). The character ` (accent grave) and the string $((dollar+open bracket) may only be specified if they are enclosed in '...' (single quotes) or if they are specified directly after a backslash ("\").

The following strings may not be specified:
– .. (two dots)
– .\ (dot + backslash)
– .' (dot + single quote)
This makes it impossible to navigate to higher-level directories.

filename prefix can be up to 511 characters in length.

Special cases

- You must specify a file name or file name prefix which starts with the string "|ftexecsv␣" for FTAC profiles which are to be used exclusively for the *ftexec* command.
  If a command prefix is also to be defined, you must specify it as follows:

  -fnp="|ftexecsv␣-p=*command prefix***"**
  (e.g.: `-fnp="|ftexecsv␣-p=\"ftshwr␣\""`)

  The same restrictions apply to the command string of the *ftexec* call as to the filename prefix during preprocessing and postprocessing.

- For FTAC profiles that are only to be used for getting monitoring data, specify the filename prefix
  "|*FTMONITOR ". The functions of the profile must permit File Preprocessing (*-ff=tp*). For details, see Example 3 on page 189.

-fnp not specified
FTAC adds no prefix to the file name.

-ls= | **-ls=@n** | **-ls=**command1

-ls specifies follow-up processing which is to be performed under your login name in the event that file transfer is successful. If *-ls* is specified, no success follow-up processing may be requested in the FT request. Specifying *-ls* only makes sense if you also make an entry for *-lf* (see below) to preclude the possibility than an intentionally unsuccessful request can circumvent the *-ls* entry. If you have defined a prefix for the file name with *-fnp* and plan follow-up processing for this file, you must specify the complete file name here.

@n for *command1*

If *-ls=@n* is specified, no success follow-up processing is permitted in the event of a successful file transfer.

-ls not specified (or **-ls=**)

does not restrict follow-up processing in the local system in the event of successful file transfer (however, see also *-lsp* or *-lss*).

-lsp=command2

-lsp defines a prefix for follow-up processing in the local system in the event of successful file transfer. FTAC then adds the character string *command2* to the follow-up processing specified in the FT request and attempts to execute the resulting command.
For example, if this option is specified as *-lsp='lpr␣'* and the request specifies *file-name* as follow-up processing, FTAC executes *lpr␣file-name* as follow-up processing.

Prefix and suffix and follow-up processing command must together not be longer than 1000 characters.

Please also bear in mind the information provided on the *-ls* option!

If a prefix was defined with *-lsp*, the character set available for specifying follow-up processing in the FT request is restricted to:
- alphanumeric characters (letters and digits)
- the special characters + = / ! ␣ - , @ _ " $ '
- a period (.) between alphanumeric characters

-lsp not specified

FTAC adds no prefix to the follow-up processing specified in the request in the event of successful file transfer.

-lss=command3

-lss defines a suffix for follow-up processing in the local system in the event of successful file transfer. FTAC then appends the character string *command3* to the follow-up processing specified in the FT request and attempts to execute the resulting command.
For example, if this option is specified as *-lss=␣file-name* and the request specifies *lpr* as follow-up processing, FTAC executes *lpr␣file-name* as follow-up processing.

Prefix and suffix and follow-up processing command must together not be longer than 1000 characters.

Please also bear in mind the information provided on the *-ls* option!

If a suffix was defined with *-lss*, the character set available for specifying follow-up processing in the FT request is restricted to:
– alphanumeric characters (letters and digits)
– the special characters + = / ! ␣ - , @ _ " $ ’
– a period (.) between alphanumeric characters

-lss not specified

FTAC adds no suffix to the follow-up processing specified in the request in the event of successful file transfer.

-lf=command4 | **@n**

-lf specifies follow-up processing to be executed under your login name if the file transfer is aborted due to an error. If *-lf* is specified, no failure follow-up processing may be requested in the FT request. Making an *-lf* entry only makes sense if you also make an entry for *-ls* (see above) to preclude the possibility that a successful request can circumvent the *-lf* entry. If you have defined a prefix for the file name with *-fnp* and plan follow-up processing for this file, you must specify the complete file name here.

@n for *command4*

If *-lf=@n* is specified, no failure follow-up processing is then permitted in the event of unsuccessful file transfer.

-lf not specified

does not restrict follow-up processing in the local system in the event of unsuccessful file transfer (Exception see *-lfp* or *-lfs*).

-lfp=command5

-lfp defines a prefix for follow-up processing in the local system in the event of unsuccessful file transfer. FTAC then sets the character string *command5* in front of the follow-up processing specified in the FT request and attempts to execute the resulting command.
For example, if this option is specified as *-lfp='lpr␣'* and the request specifies *file-name* as follow-up processing, FTAC executes *lpr␣file-name* as follow-up processing.
Prefix and suffix and follow-up processing command must together not be longer than 1000 characters.

Please also bear in mind the information provided on the *-lf* option!

If a suffix was defined with *-lfs*, the character set available for specifying follow-up processing in the FT request is restricted to:
– alphanumeric characters (letters and digits)
– the special characters + = / ! ␣ - , @ _ " $ '
– a period (.) between alphanumeric characters

-lfp not specified

FTAC sets no prefix in front of the follow-up processing specified in the request in the event of unsuccessful file transfer.

-lfs=command6

-lfs defines a suffix for follow-up processing in the local system in the event of unsuccessful file transfer. FTAC then sets the character string *command6* after the follow-up processing specified in the FT request and attempts to execute the resulting command.
For example, if this option is specified as *-lfs=␣file-name* and the request specifies *lpr* as follow-up processing, FTAC executes *lpr␣file-name* as follow-up processing.

Prefix and suffix and follow-up processing command must together not be longer than 1000 characters.

Please also bear in mind the information provided on the *-lf* option!

If a suffix was defined with *-lfs*, the character set available for specifying follow-up processing in the FT request is restricted to:
– alphanumeric characters (letters and digits)
– the special characters + = / ! ␣ - , @ _ " $ '
– a period (.) between alphanumeric characters

-lfs not specified

FTAC sets no suffix after the follow-up processing specified in the request in the event of unsuccessful file transfer.

-wm=o | **-wm=n** | **-wm=e** | **-wm=one**

-wm specifies which write modes may be used in the file transfer request and what they effect.

o (overwrite) In the FT request of openFT or FTAM partners, only *-o* or *-e* may be entered for write mode. The receive file is overwritten if it already exists, and is created if it does not yet exist.

With FTP partners, *-n* may also be entered if the file does not yet exist.

n (no overwrite) In the FT request *-o*, *-n* or *-e* may be entered for write mode.
The receive file is created if it does not yet exist. If the receive file already exists, the request is not executed.

e (extend) In the FT request only *-e* may be entered for write mode, i.e. the receive file is extended by appending the transferred file to the end if the receive already exists. The receive file is created if it does not yet exist.

one (default value)
means that the FT profile does not restrict the write mode.

-c=y | **-c=n**

Using *-c*, you can determine whether data encryption is required or forbidden. If the setting in the profile does not correspond to the setting in the request, the request is denied. The setting is not valid for file management requests, since there is no encryption for these requests.

y Only requests *with* data encryption may be processed using this profile.

n Only requests *without* data encryption may be processed using this profile.

-c not specified
Data encryption is neither required nor forbidden.

-txt=text
enables you to store a comment in the FT profile (up to 100 characters).

-txt not specified
the FT profile is stored without a comment.

CAUTION

If you use the options *-ff=p*, *-fn*, *-fnp*, *-ls* , *-lsp*, *-lss*, *-lf*, *-lfp* or *-lfs*, you must remember

– that a file-name restriction can be bypassed by renaming the file unless follow-up processing is also restricted;

– that follow-up processing must always be restricted for both successful and unsuccessful file transfer and, if necessary, equivalent restrictions must exist for any permitted preprocessing;

– that prefixes for the file name and follow-up processing must be matched to one another;

– that no symbolic links should occur in the part of your file tree that is referenced by the file name prefix.

– that restrictions applied to preprocessing, postprocessing, or follow-up processing can be circumvented if it is possible to replace this command with, for example, a "Trojan horse".

Example

1. You wish to create an FT profile for the following purpose:

   The Duck Goldmines are to be able to send their monthly reports from their computer *goldmine* to the president at head office via file transfer. The file *monthlyreport_goldmine01* is to be printed out after transfer. The command required to create such an FT profile at head office is:

   ```
   ftcrep␣goldmrep␣fortheboss␣-d=20171231␣-dir=f\
   ␣-pn=goldmine␣-fn=monthlyreport_goldmine01\
   ␣-ls='lpr␣monthlyreport_goldmine01'␣-lf=@n␣-wm=o
   ```

   The FT profile has the name *goldmrep* and the transfer admission *fortheboss*. It permits only the *monthlyreport_goldmine01* file to be transferred to the bank. Following successful transfer, the file is printed out in the bank. Follow-up processing after unsuccessful file transfer is, however, prohibited. The transfer admission is only valid until December 30, 2017, the FT profile disabled as of 00:00 hours on December 31, 2017.

2. You want to set up the standard admission profile on your user ID in such a way that only the file transfer and file creation functions are possible. This profile can, for instance, be used by FTAM partners that always have to specify the user ID and the password for inbound access.

   The command is as follows:

   ```
   ftcrep␣@s␣@n␣-wm=n␣-ff=t
   ```

3. You want to define an admission profile *monitor1* that only allows monitoring data to be output. Assign *onlyftmonitor* as the transfer admission. The command is as follows:

   ```
   ftcrep monitor1 onlyftmonitor -ff=tp -fnp="|*FTMONITOR "
   ```

   The purpose of the blank after *FTMONITOR is to automatically separate any options specified during the call from the command. A profile such as this can be used to call the openFT monitor (e.g. using the *ftmonitor* command) and in the *ncopy* command. The admission profile is only valid for communicating via the openFT protocol.

   You will find further details under "Monitoring with openFT" in the openFT manual "Installation and Administration".

5.8 ftdel - Delete a file in a remote system

With *ftdel* you can delete files in the remote system.

Format

ftdel -h |
<partner 1..200>![<file name 1..512>]
[<transfer admission 8..67> | @n | @d |
<user ID 1..67>[,[<account 1..64>][,[<password 1..64>]]]]
[-p=[<management password 1..64>]]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

partner**!**file name
Specifies which file in which remote system has to be deleted.

partner
Partner is the name of the partner system in the partner list or the address of the partner system. For details on address specifications, see section “Defining the partner computer” on page 90.

file name
file name can be either absolute or relative to the remote login admission. If the file name in the remote system has been predefined in an FT profile, it must not be specified here.

If the partner system is running openFT for BS2000/OSD, elements from PLAM libraries may also be specified here (Syntax: Libname/Element type/Element name).

transfer admission | **@n** | **@d** |
user ID[**,**[account][**,**[password]]]
In order to execute file management requests in the remote system, you must furnish the remote system with proof of identity. For this purpose, you will need login admission in the syntax valid for the remote system. You can specify this transfer admission

– as an FTAC transfer admission if FTAC is used in the remote system,

– or as a login/LOGON admission in the syntax used by the remote system (user ID, possibly together with account or password).

For details, see section “Transfer admission” on page 94.

@n for *transfer admission*
: By entering *@n* you specify that the remote system requires no login admission.

@d for *transfer admission*
: Specifying *@d* (blanked transfer admission) causes openFT to query the transfer admission on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the transfer admission.

A binary password or binary transfer admission must be specified in hexadecimal format in the form x'...' or X'...'. If you enter the password directly, remember to insert a backslash (\) to escape the single quotes if you did not enclose the remote login admission in double quotes, for example: X\'c6d9e4c5\'.

password not specified
: Omitting the password necessary for admission causes openFT to query the password on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the password. In this case, single quotes must not be escaped by a backslash (\).

  Nevertheless, you have to specify the commas, e.g.:

```
ftdel file partner!file user-id,,
```
  or
```
ftdel file partner!file user-id,account,
```

neither *transfer admission* nor *user ID* specified
: causes the same as *@d*, i.e. openFT queries the transfer admission on the screen after the command is entered. Your (blanked) entry is always interpreted as transfer admission and not as user ID.

-p=[management-password]
: If the file in the remote system is protected by a password, you must enter this password here.

  A binary password must be entered in hexadecimal form x'...' or X'...'. This is of relevance for links to openFT for BS2000/OSD, because BS2000 supports the definition of hexadecimal passwords. If you enter the password directly, remember to insert a backslash (\) to escape the single quotes, for example: X\'c6d9e4c5\'.

management password not specified

Specifying *-p=* causes openFT to query the password on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the password. In this case, single quotes must not be escaped by a backslash (\).

Example

The file *junk* in the BS2000 computer *bs2r1* under login name *jim* with account number *a1234ft* and password *C'pwd'* is to be deleted from your system. The file is protected by the password *abcd*.

```
ftdel␣bs2r1!junk␣jim,a1234ft,C\'pwd\'␣-p=C\'abcd\'
```

5.9 ftdeldir - Delete remote directories

You can use *ftdeldir* to delete a directory on a remote system. For this to be possible, the remote system must support this function.

You can only delete remote directories.

Format

ftdeldir -h |
<partner 1..200>![<file name 1..512>]
[<transfer admission 8..67> | @n | @d |
<user ID 1..67>[,[<account 1..64>][,[<password 1..64>]]]]
[-p=[<management password 1..64>]]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

partner**!**[file name]
Specifies what directory is to be deleted on what computer.

partner
Partner is the name of the partner system in the partner list or the address of the partner system. For details on address specifications, see section “Defining the partner computer” on page 90.

file name
Name of the directory that is to be deleted.
You can specify *file name* absolutely or relative to the remote login authorization. If the file name in the remote system is predefined by an admission profile then it may not be specified here.

If openFT for BS2000/OSD is running on a partner system then an empty PLAM can be specified here. This deletes the PLAM library.

If the directory or PLAM library is not empty then you can delete the files or elements with *ftdel* before calling *ftdeldir*.

transfer admission | **@n** | **@d** |
user ID[**,**[account][**,**[password]]]

Before you can modify the attributes of a file on a remote system, you must first identify yourself at the system. To do this, you need an authorization in the syntax used at the remote system. You can specify this transfer admission

– as an FTAC transfer admission if FTAC is used in the remote system,

– or as a login/LOGON admission in the syntax used by the remote system (user ID, possibly together with account or password).

For details, see section “Transfer admission” on page 94.

@n for *transfer admission*

By entering *@n* you specify that the remote system requires no login admission.

@d for *transfer admission*

If you specify *@d* (blanked) then the transfer admission is queried on the screen after the command has been sent. Your input is invisible to prevent unauthorized persons from seeing the transfer admission.

A binary password or binary transfer admission must be specified in hexadecimal format in the form x'...' or X'...'. If you enter the password directly, remember to invalidate the single quotes with a backslash (\) unless you have enclosed the remote login authorization in double quotes, for example X\'c6d9e4c5\'.

password not specified

If you omit a password which is required for authorization then it is queried on the screen after the command has been sent. Your input is invisible to prevent unauthorized persons from seeing the password. In this case, quotes must not be invalidated with a backslash (\).

Please note that you still have to enter the commas, for example:

```
ftdeldir partner!file identification,,
```

or

```
ftdeldir partner!file identification,account,
```

neither *transfer admission* nor *user ID* specified

This has the same effect as *@d*, i.e. the transfer admission is queried on the screen after the command has been sent. openFT always interprets your (hidden) input as a transfer admission and not as a user ID.

-p=[management password]

If the directory is protected by a password in the remote system then you must specify this here.

The password must be specified in hexadecimal format in the form x'...' or X'...'. This is of relevance in the case of a connection with openFT for BS2000/OSD since it is possible to define hexadecimal passwords in BS2000. If you enter the password directly, remember to invalidate the single quotes with a backslash (\), for example: -p=X\'c6d9e4c5\'.

management password not specified

If you specify *-p=* then the password is queried on screen after the command has been sent. Your input is invisible to prevent unauthorized persons from seeing the password. In this case, quotes must not be invalidated with a backslash (\).

5.10 ftdelp - Delete FT profiles

ftdelp stands for "delete profile". You should occasionally thin out the set of profiles (with *ftshwp*) to ensure that no out-of-date admission profiles are retained that could potentially threaten the security of your system.

Format

ftdelp -h |
<profile name 1..8> | @s | @a
[-s=[<transfer admission 8..32> | @a | @n]
[,<user ID 1..32> | @a | @adm]]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

profile name | **@s** | **@a**
is the name of the FT profile you wish to delete.

@s for *profile name*
Deletes the standard admission profile for the user ID.

@a for *profile name*
profile name is not used as a criterion for selecting the FT profile to be deleted. If you do not identify the profile more closely with *-s* (see below) you will delete all of your FT profiles.

-s=[transfer admission | **@a** | **@n**][**,**user ID | **@a**]
-s is used to specify criteria for selecting the FT profiles to be deleted.

transfer admission
is the transfer admission of the FT profile to be deleted. A binary transfer admission must be specified in the form x\'...\' or X\'...\' .

@a for *transfer admission*
deletes either the FT profile specified by *profile name* (see above) or all of your FT profiles.

@n for *transfer admission*
deletes FT profiles with no transfer admissions.

transfer admission not specified
causes to query the transfer admission on the screen after the command is entered. Your entry is not displayed to prevent

unauthorized persons from seeing the transfer admission. To exclude the possibility of typing errors, the program prompts you to enter the transfer admission a second time. If you just press <ENTER>, this has the same effect as specifying *@a*.

,user ID
: As user, you can enter only your own login name here.

@a for *user ID*
: allows you to delete FT profiles belonging your own login name.

@adm for *user ID*
: For the FTAC and ADM administrator only.

user ID not specified
: deletes only profiles belonging to the user's own login name, regardless of who issues the command.

-s not specified
: if *@a* is specified for *profile name*, all the FT profiles belonging to the login name under which the *ftdelp* command is issued are deleted. Otherwise, the FT profile with the specified name is deleted.

Example

The FT profile *goldmrep* is to be deleted.

```
ftdelp␣goldmrep
```

5.11 ftedit - Load local or remote files in the openFT editor

The shell command *ftedit* allows you to load local or remote files in the openFT editor.

Please note that you require a graphics-capable terminal in order to use the *ftedit* command.

Format

ftedit [-ro]
[-n=<line>]
[-t |- b |-u]
[-ccs=<ccs>]
[-tad=<tad> <partner>!]<file>

Description

-ro Loads the file in write-protected mode. You can only read the file. This corresponds to the “View” function in the Explorer interface.

-n=line
The editor window is positioned on the specified line after the file is loaded.

-t | **-b** | **-u**

In the case of remote files, the file type to be used when the file is transferred to openFT.

-t (default value for openFT partners)
The file contains text with variable record lengths. Records are terminated by the newline character \n.

Maximum record length = 32767 bytes

-u The file contains variable record length binary data structured by the user. Every record starts with 2 bytes that specify the length of the record.

Maximum record length = 32767 bytes

-b The file contains an unstructured sequence of binary data. If you specify the option *-b* together with *-r* (maximum record length), the file contains binary data with the record length specified under *-r*. The size of the send file must then be a multiple of this record length.

-ccs=ccs

Name of the character set that is to be used on opening the file. For more information, see section “Code tables and coded character sets (CCS)” on page 85.

Default: the character set defined as the default in the local openFT system.

-tad=tad

Transfer admission in the partner system in the case of remote files.

You can specify the transfer admission

– as an FTAC transfer admission if FTAC is used in the remote system,
– or as a login/LOGON admission using the syntax of the remote system (user ID, where necessary with account and/or password).

You will find further details in the section “Transfer admission” on page 94.

partner

For remote files it is necessary to specify an openFT partner name.

Partner is the name of the partner system in the partner list or the address of the partner system. For details on address specifications, see section “Defining the partner computer” on page 90.

file Name of the file to be loaded in the openFT editor.

You can specify an absolute path or a relative path for the file name with a maximum length of 512 characters. Please note that the maximum lengths of file names are system-dependent; for example, in Unix systems it is 512 and in Windows systems a maximum of 256 characters. If the file name contains blanks, you must enclose it in double quotes (e.g. "file name"). If the remote partner requires single quotes around the file name, unlike at the shell level you do not have to invalidate these (e.g. ’file name’).

5.12 ftexec - Execute operating system commands in remote system

The *ftexec* command is used to execute operating system commands in the remote system. The resulting output for *stdout* and *stderr* is output in the local system on standard output (*stdout*) or standard error (*stderr*).

ftexec is only available for openFT partners, FTP partners and FTAM partners from Fujitsu Technology Solutions.

The end status, i.e. the result of the command, is also output in the local system as the end status of the *ftexec* command. If the end status received exceeds the value range of the local end status (Unix systems have only a 1-byte end status while Windows systems have a 4-byte end status), only the contents of the least significant byte are output.

If the command is not executed in the remote system, an end message from the *ftexec* command is output to *stderr*, and *ftexec* terminates with the end status 255. The end status has a system-specific meaning.

For output operations to *stdout*, it is possible to define character sets (*-lc*, *-rc*). For output operations to *stderr*, the following character sets are used depending on the system:
– BS2000/OSD, z/OS: character set defined in the system
– Unix systems: ISO8859-1
– Windows systems: CP850

You will find further information on creating FTAC profiles for the *ftexec* function in the description of the *ftcrep* command, in particular the *-fnp* option on page 182.

Format

```
ftexec   -h |
         [ -t | -b | -l ]
         [ -c ]
         [ -lc=<CCS name 1..8> ] [ -rc=<CCS name 1..8> ]
           <partner 1..200>
           <command> | -
         [ <transfer admission 8..67> | @n | @d |
           <user ID 1..67>[,[<account 1..64>][,[<password 1..64>]]] ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-t This option indicates the transfer format for *stdout* is text. Tabulator expansion is deactivated. Default value if a CCS name is specified (*-lc* and/or *-rc*).

-b This option indicates that the transfer format for *stdout* is binary without conversion. Default value if no CCS name is specified (neither *-lc* nor *-rc*).

-l This option indicates that the transfer format for *stdout* is binary with <CRLF> converted to <LF> (transfer of text in binary format). This mode is only of use if both partners use ISO 646 or ISO8859-1 as the text format.

-c Specifies that the data is also to be encrypted at transfer. The encryption of the request description data is not affected by this option. If the partner system cannot work with encryption, the request is rejected.

-lc=CCS name

(local coding) specifies the type of coding (character set) to be used to read the local file. *CCS name* must be known in the local system.

The default value is the character set defined by the FT administrator.

-lc may not be combined with *-b* or *-l*.

Details about the CCS name and the associated code tables can be found in section “Code tables and coded character sets (CCS)” on page 85.

-rc=CCS name

(remote coding) specifies the type of coding to be used to read the data at the standard output from the remote command. *CCS name* must be known in the remote system.

The default value is the character set defined in the remote system.

-rc may not be combined with *-b* or *-l*.

The option *-rc* is supported only by the openFT protocol and partners with openFT V10.0 or higher. Please note that not all partner systems support all the character sets that are possible in the local system.

partner

partner is the name of the partner system in the partner list or the address of the partner system. For details on address specifications, see section “Defining the partner computer” on page 90.

command | **-**

command is the command to be executed in the remote system. The syntax and the processing of the statements and commands depend on the conventions of the system on which the command is to be executed. A command sequence can only be processed in the remote system if an FT product that supports this function is being used there.
The maximum length of the command depends on the maximum length of the file names in the remote partner and the number of special characters in the command itself. With the current restriction of the length of a file name to 512 characters, the command can have a maximum of 478 characters. Special characters count as being two characters.

- (dash) for *command*

You must enter the command after sending the *ftexec* command via *stdin*. You terminate entry by pressing <END> or CTRL+D.

transfer admission | **@n** | **@d** |
user ID[**,**[account][**,**[password]]]

If you want to execute a command on a remote system, you must furnish the remote system with proof of identity. For this purpose, you will need login admission in the syntax valid for the remote system. You can specify this transfer admission

– as an FTAC transfer admission if FTAC is used in the remote system,

– or as a login/LOGON admission in the syntax used by the remote system (user ID, possibly together with account or password).

For details, see section “Transfer admission” on page 94.

@n for *transfer admission*

By entering *@n* you specify that the remote system requires no login admission.

@d for *transfer admission*

Specifying *@d* (blanked transfer admission) causes openFT to query the transfer admission on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the transfer admission.

A binary password or binary transfer admission must be specified in hexadecimal format in the form x'...' or X'...'. If you enter the password directly, remember to insert a backslash (\) to escape the single quotes if you did not enclose the remote login admission in double quotes, for example: X\'c6d9e4c5\'.

password not specified
: Omitting the password necessary for admission causes openFT to query the password on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the password. In this case, single quotes must not be escaped by a backslash (\).

  Nevertheless, you have to specify the commas, e.g.:

  ```
  ftexec system command user-id,,
  ```
  or
  ```
  ftexec system command user-id,account,
  ```

neither *transfer admission* nor *user ID* specified
: causes the same as *@d*, i.e. openFT queries the transfer admission on the screen after the command is entered. Your (blanked) entry is always interpreted as transfer admission and not as user ID.

Example

1. You want to look at the last 12 log records in the remote Unix system *ux1* using the transfer admission *Transuser1*:

   ```
   ftexec␣ux1␣“ftshwl␣-nb=12“␣Transuser1
   ```

2. You want to look at the last 12 log records in the remote BS2000 system *bs2* using the transfer admission *Transbs2*:

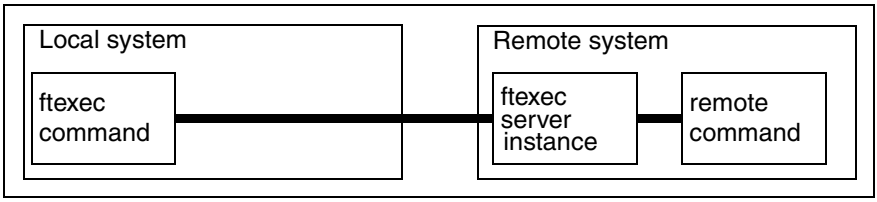
   ```
   ftexec␣-t␣bs2␣"/SH-FT-LOG␣,12"␣Transbs2
   ```

3. You want to look at the last 12 log records in the remote z/OS system *zos1* using the transfer admission *TranszOS*:

   ```
   ftexec␣-t␣zos1␣"ftshwlog␣,12"␣TranszOS
   ```

5.12.1 Messages from the ftexec command

Several openFT components in the local and remote systems participate in the execution of an *ftexec* command. Any of these instances can be responsible for the messages issued during execution:



In the local system, this is the specified *ftexec* command which is very similar to the *ncopy* command in its execution. Consequently, all the *ncopy* command messages may occur, the only difference being that they start with *ftexec*.

In the remote system, both the remote command itself and the *ftexec* server which monitors the execution of the remote command may handle requests. However, messages from the *ftexec* server are mapped to *ncopy* command messages wherever possible, i.e.:

- If the end status for *ftexec* is not 255, then all *stderr* output originates from the command executed in the remote system (depending on the remote command involved). Messages from the other instances are only possible if the end status is 255.
- Messages from the *ftexec* command responsible for the transfer of data can have another additional meaning:

  Request *request ID*: Remote system: Error in pre-/postprocessing

  Request *request ID*: Remote system: Exitcode *code* from pre-/postprocessing

  Meaning:
  The local preprocessing command could not be executed successfully. The exit code here is the exit code of the *ftexec* server, i.e. 255.

  Request *request ID*: Remote system: Transfer admission invalid

  Other possible meaning:
  The transfer admission does not permit any command execution.

  Request *request ID*: Remote system: Syntax error in resulting file name.

  Other possible meaning:
  The command string is too long for the remote partner.

Request *request ID*: Remote system: File/directory'*file*' not found

Other possible meaning:
The file name prefix in the remote FTAC profile does not start with "|ftexecsv␣".

ftexec: Invalid parameter 'c'

Meaning:
Encryption of user data is not enabled.

- Messages deriving from *ftexec* server instance messages (these start with "ftexecsv:"):

  Request *request ID*: Remote system: File/directory does not exist

  Meaning:
  The command specified in *ftexec* does not exist in the remote system - at least not under the explicitly specified or implicitly assumed path. If the partner is a Unix system, this message can also mean that the file exists but cannot be executed as a command or that a resource bottleneck occurred when an attempt was made to start the command.

  Request *request ID*: Remote system: Access to ... denied

  Meaning:
  The command specified in *ftexec* is not an executable command or includes invalid characters (see *ftcrep* command, *-fnp* option on page 182).

  Request *request ID*: Remote system: Resource bottleneck

  Meaning:
  A resource bottleneck occurred when an attempt was made to start the command specified in *ftexec*.

  Request *request ID*: Remote system: File structure error

  Meaning:
  – An error occurred while reading the *stdout* or *stderr* data generated when the remote command was executed.
  – A record created by the command specified in *ftexec* cannot be entered in the *ftexec* server buffer. An attempt was probably made to read pure binary output as text.
  – The *ftexec* server received an error flag while forwarding the data from the remote command to the openFT server.

Request *request ID*: Internal error. Error code *err_code*

Meaning:
An internal error occurred in the remote *ftexec* server.

- Messages from the *ftexec* command itself (these start with "ftexec:"):

  Request *request ID*: File structure error

  Meaning:
  The data received does not correspond to the *ftexec* format. It may originate from a remote file or from normal preprocessing. Check whether the appropriate transfer admission has been selected.

  Internal error. Error code *err_code*

  Meaning:
  An internal error *err_code* occurred during the processing of the *ftexec* command.

5.13 fthelp - Display information on the log record reason codes

With *fthelp*, you can have the meanings of the reason codes for the log function displayed on the screen (RC column in *ftshwl* output).

You can also request the output of the message texts associated with the exit codes of certain FT commands.

Format

fthelp -h | <number 1..ffff>

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

number

This is a four-digit reason code from the log function or the exit code of an FT command belonging to a synchronous FT request. The reason code contains encoded information on an FT request accepted by openFT.

The reason codes and their meanings are listed in the section “Reason codes of the logging function” on page 288.

The exit codes (= message numbers) are listed in section “openFT messages” on page 378.

Example

You wish to find out the meaning of reason code 3001.

```
fthelp␣3001
3001 Request rejected. Invalid user identification.
```

Thus, reason code 3001 means that the specified login name or transfer admission is invalid.

5.14 ftinfo - Output information on the openFT system

ftinfo outputs information about the installed openFT system.

Format

ftinfo -h
[-csv]

Output

ftinfo always outputs the values in CSV format even if the *-csv* option is not specified:

Name	Type	Values
CmdUiVer	Number	Version of the User Command Interface, e.g. 1100 for V11.00. The User Command Interface provides the user and administrator commands.
CmdTiVer	Number	Version of the Tool Command Interface, e.g. 100 for V1.00.
OsType	String	Name of the operating system: Windows, Unix, BS2000/OSD, z/OS.
UserId	String	Current (calling) user ID.
IsFtAdm	Number	1 for UserId=FT administrator, 0 otherwise
IsFtacAdm	Number	1 for UserId=FTAC administrator, 0 otherwise.
FtLang	String	Set language: de (German), en (English).
FtacAccess	String	Access rights to FTAC files, displayed only to ensure compatibility with predecessor versions.
CcsName	String	CCS name of the character set currently defined in openFT.
Home	String	Home directory of the calling user ID.

Name	Type	Values
Limited	String	*NO or YYYY-MM-DD *NO: The installed openFT product is NOT a limited period evaluation version. YYYY-MM-DD: The installed openFT product is a limited period evaluation version that can be used until the specified date. openFT can no longer be used after the date displayed.
IsAdmAdm	Number	1 for UserId=ADM administrator, 0 otherwise

Example

```
ftinfo

CmdUiVer;CmdTiVer;OsType;UserId;IsFtAdm;IsFtacAdm;FtLang;CcsName
;Home;Limited;IsAdmAdm

1100;100;"Unix";"admin";1;1;"en";"ISO88591";"/home/usr/admin";
*NO;0
```

5.15 ftmod - Modify file attributes in a remote system

With *ftmod* you can modify the attributes of a file in a remote system. Depending on the partner (openFT, FTAM or FTP), the following file attributes can be modified:

With openFT partners:

– File name
– Access rights (not if the partner system is a Windows system)

With FTAM partners:

– File name
– Access rights (not if the partner system is a Windows system)
– Availability of the file
– Account for file storage costs
– Legal stipulation on using the file
– Future file size

With FTP partners:

– File name

Format

```
ftmod -h |
        <partner 1..200>![<file name 1..512>]
      [ <transfer admission 8..67> | @n | @d |
        <user ID 1..67>[,[<account 1..64>][,[<password 1..64>]]] ]
      [ -p=[<management password 1..64>] ]
      [ -nf=<new file name 1..512> ]
      [ -av=i | -av=d ]
      [ -ac=<new account 1..64> ]
      [ -fs=<future filesize 1..2**63-1> ]
      [ -am=[+][r][i][p][x][e][a][c][d] | -am=@rw | -am=@ro ]
      [ -lq=<legal qualification 1..80> ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

partner**!**[file name]
Specifies for which file and on which system the attributes are to be modified.

partner
partner is the name of the partner system in the partner list or the address of the partner system. For details on address specifications, see section “Defining the partner computer” on page 90.

file name
file name can be either absolute or relative to the remote login admission. If the file name in the remote system has been predefined in an FT profile, it must not be specified here.

If the partner system is running openFT for BS2000, elements from PLAM libraries may also be specified here (Syntax: Libname/Element type/Element name).

transfer admission | **@n** | **@d** |
user ID[**,**[account][**,**[password]]]
In order to modify the file attributes in the remote system, you must furnish the remote system with proof of identity. For this purpose, you will need login admission in the syntax valid for the remote system. You can specify this transfer admission

– as an FTAC transfer admission if FTAC is used in the remote system,

– or as a login/LOGON admission in the syntax used by the remote system (user ID, possibly together with account or password).

For details, see section “Transfer admission” on page 94.

@n for *transfer admission*
By entering *@n* you specify that the remote system requires no login admission.

@d for *transfer admission*
Specifying *@d* (blanked transfer admission) causes openFT to query the transfer admission on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the transfer admission.

A binary password or binary transfer admission must be specified in hexadecimal format in the form x'...' or X'...'. If you enter the password directly, remember to insert a backslash (\) to escape the single quotes if you did not enclose the remote login admission in double quotes, for example: X\'c6d9e4c5\'.

password not specified
: Omitting the password necessary for admission causes openFT to query the password on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the password. In this case, single quotes must not be escaped by a backslash (\).

  Nevertheless, you have to specify the commas, e.g.:

  ```
  ftmod partner!file user-id,,
  ```
  or
  ```
  ftmod partner!file user-id,account,
  ```

neither *transfer admission* nor *user ID* specified
: causes the same as *@d*, i.e. openFT queries the transfer admission on the screen after the command is entered. Your (blanked) entry is always interpreted as transfer admission and not as user ID.

-p=[management password]
: If the file in the remote system is protected by a password, you must enter this password here.

  A binary password must be entered in hexadecimal form x'...' or X'...'. This is of relevance for links to openFT for BS2000/OSD, because BS2000 supports the definition of hexadecimal passwords. If you enter the password directly, remember to insert a backslash (\) to escape the single quotes if you did not enclose the remote login admission in double quotes, for example: X\'c6d9e4c5\'.

  management password not specified
  : Specifying *-p=* causes openFT to query the password on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the password. In this case, single quotes must not be escaped by a backslash (\).

-nf=new file name
: This indicates the new name for the file *file name* in the partner system. The name *file name* is then no longer valid. *new file name* can be either absolute or relative to the remote login admission.

-nf not specified
: The file name remains unchanged.

-av=i | **-av=d**
: Indicates the availability of the file in an FTAM partner system. This parameter can have one of two values: *immediate* or *deferred*. A file may be *deferred* if it has been archived, for example. The partner is responsible for interpreting the term *deferred*. The FTAM partner conventions must therefore be observed here.

  The following values are possible:

  i In the remote system, the file attribute is set to *immediate*.

  d In the remote system, the file attribute is set to *deferred*. The file on the partner system can then be placed in an archive, for example.

  Requests involving openFT or FTAM partners that do not support the storage group are rejected.

-av not specified
: The previous value for availability remains unchanged.

-ac=new account
: With FTAM partners, this indicates the number of the account to which file storage fees are to be charged. This parameter must be set in accordance with partner system conventions.

  Requests involving openFT or FTAM partners that do not support the storage group are rejected.

-ac not specified
: The previous account number remains unchanged.

-fs=future filesize
: With FTAM partners, this indicates the expected file size. This is used as a guide for system-specific optimization.

  Requests involving openFT or FTAM partners that do not support the storage group are rejected.

-fs not specified
: The previous file size remains unchanged.

-am=[**+**][**r**][**i**][**p**][**x**][**e**][**a**][**c**][**d**] | **@rw** | **@ro**

This changes the access rights for a file in the remote system. Old access rights can also be replaced with new ones.

The following values can be specified for the *-am* parameter: *+*, *r*, *i*, *p*, *x*, *e*, *a*, *c*, *d* or any combination of these values as well as *@rw*, or *@ro*.

+ with FTAM partners means that the file receives a new set of access rights in addition to the existing rights. This entry is only relevant for FTAM partners that support more than one set of access rights.

+ not specified
the existing access rights of the file in the remote system are replaced by the specified access rights.

r means that the file can be read.

r not specified
The file cannot be read.

i with FTAM partners means that data units, such as records, can be inserted in the file.

i not specified
No data units can be inserted in the file.

p means that the file can be overwritten.

p not specified
The file cannot be overwritten.

x means that data can be appended to the file.

x not specified
The file cannot be extended.

e with FTAM partners means that data units, such as records, can be deleted from the file.

e not specified
No data units can be deleted from the file.

a means that the file attributes can be read.

a not specified
The file attributes cannot be read.

c means that the file attributes can be changed.

c not specified
The file attributes cannot be changed.

d means that the file can be deleted.

d not specified
The file cannot be deleted.

@rw
is the short form of the common access rights *read-write* (*rpxeacd*), and thus simplifies input.

@ro
is the short form of the common access rights *read-only* (*rac*), and thus simplifies input.

If the partner system is a Windows system, you cannot change the access rights of the destination file.

With Unix or BS2000 partner systems, only the following access rights can be set for a file:

Access mode	Short form	Unix system	BS2000	Access rights
rpxeacd	@rw	rw*	ACCESS=WRITE	read-write
rac	@ro	r-*	ACCESS=READ	read-only
pxeacd		-w*	only with BASIC-ACL (Access Control List)	write-only
ac		--*	only with BASIC-ACL (Access Control List)	none

* The x bit is not changed by *ftmod*.

Requests involving FTP partners or involving FTAM partners that do not support the security group are rejected.

-am not specified
The current access rights remain unchanged.

-lq=legal qualification

With FTAM partners, this specifies a legal qualification for the file (similar to a copyright). This may not exceed 80 characters.

Requests involving openFT or FTAM partners that do not support the security group are rejected.

-lq not specified

The current legal qualifications remain unchanged.

Example

You wish to reset the access rights of the remote file *junk* from *read-only* to *read-write*. The file is on the BS2000 computer *bs2r1* under login name *jim* with account number *a1234ft* and password *C'pwd'*. The file is protected by the password *abcd*.

```
ftmod␣bs2r1!junk␣jim,a1234ft,C\'pwd\'␣-p=C'abcd'␣-am=@rw
```

5.16 ftmoda - Modify admission sets

ftmoda stands for "modify admission set".

When *ftmoda* is issued by an FTAC user, it modifies one or more of the settings for basic functions in that user's admission set (MAX. USER LEVELS).

You can assign a security level of between 0 and 100 for each basic function. These values have the following meanings:

0 The basic function is locked, i.e. it is not released for any partner system.

1 to **99** The basic function is only released for partner systems with the same or a lower security level. You can use the *ftshwptn* command to display the security level of a partner system.

100 The basic function is available for all partner functions.

For basic functions, consult the table on page 219.

The meaning of the numbers in the admission set has been changed in openFT V10.0. Now, all integers between 0 and 100 are analyzed and are compared with the partner system security levels to determine whether they are smaller than or equal to these values.

Format

```
ftmoda -h |
        [ <user ID 1..32> | @s ]
        [ -priv=y ]
        [ -admpriv=y ]
        [ -ml=s | -ml=0..100 ]
        [ -os=s | -os=0..100 ]
        [ -or=s | -or=0..100 ]
        [ -is=s | -is=0..100 ]
        [ -ir=s | -ir=0..100 ]
        [ -ip=s | -ip=0..100 ]
        [ -if=s | -if=0..100 ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

user ID | **@s**

Users can enter only their own login names here. *@s* is not permitted.

user ID not specified
modifies the admission set of the login name under which *ftmoda* is entered.

-priv=y
can only be used by the FTAC administrator.

-admpriv=y
can only be used by the ADM administrator.

-ml=s | **-ml=**0..100
sets the same value for all six basic functions.

Possible values are:

s sets each of the basic functions to the value defined in the standard admission set.

0 disables all of the basic functions.

1 to **99**
All basic functions are released only for partner systems whose security level is equal to or lower than the specified value.

100 All basic functions are released for all partner systems. For outbound file management functions, no check is made.

-ml not specified
leaves the settings in the admission set unchanged if none of the following entries are made.

-os=s | **-os=**0..100
sets the value for the basic function *outbound send,* see page 219 for possible values. *outbound send* means that requests initiated in your local system send data to a remote system.

-or=s | **-or=**0..100
: sets the value for the basic function *outbound receive*, see page 219 for possible values. *outbound receive* means that requests initiated in your local system fetch data from a remote system.

-is=s | **-is=**0..100
: sets the value for the basic function *inbound send*, see page 219 for possible values. *inbound send* means that a remote partner system fetches data from your local system.

-ir=s | **-ir=**0..100
: sets the value for the basic function *inbound receive*, see page 219 for possible values. *inbound send* means that a remote partner system sends data to your local system.

-ip=s | **-ip=**0..100
: sets the value for the basic function *inbound follow-up processing* + *preprocessing* + *postprocessing*, see page 219 for possible values. This determines whether or not a remote system may request follow-up, pre- or postprocessing on your local system.

-if=s | -**if=0**..100
: sets the value for the basic function *inbound file management*, see page 219 for possible values.

  Please note that subcomponents of *inbound file management* are affected by other settings, see "Dependencies concerning inbound file management" on page 220

-os, *-or, -is, -ir, -ip* or *-if* not specified
: leaves the setting for the respective basic function unchanged.

Possible values for the basic functions

The following values are possible for the individual basic functions (*-os*, *-or, -is, -ir*, *-ip* and *-if*):

s The specifications in the default admission record apply to the basic functions.

0 The basic function is locked.
With some basic functions, this can also affect inbound file management components. For details, refer to the table on page 220.

1 to **99**
The basic function is only released for partner systems on which the security level is less than or equal to the specified value.

100 The basic function is released for all partner systems.

Dependencies concerning inbound file management

The subcomponent *Display file attributes* is controlled by the basic *function inbound send*. In addition, the following dependencies on other on other settings exist for some components:

Inbound file management function	**Values of the admission set or extension in profile**
Display file attributes	Inbound Send (IBS) enabled
Modify file attributes	Inbound Receive(IBR) **and** Inbound File Management(IBF) enabled
Rename files	Inbound Receive(IBR) **and** Inbound File Management(IBF) enabled
Delete files	Inbound Receive(IBR) enabled **and** Write mode = overwrite in profile
Display directories	Inbound File Management(IBF) enabled
Create, rename and delete directories	Inbound File Management(IBF) enabled **and** direction = from partner in profile

Example

The user Donald wishes to change the admission set for his login name *donald* to prevent remote systems accessing his login name, while still allowing to send files. This requires that the *outbound* basic functions be enabled and the *inbound* basic functions disabled. This can be achieved with the following command:

```
ftmoda␣-os=100␣-or=100␣-is=0␣-ir=0␣-ip=s␣-if=0
```

Donald specifies the value *s* for the basic function *inbound follow-up* + *preprocessing* + *postprocessing* (*-ip* option), which refer to the standard admission set, where this basic function is also disabled.

5.17 ftmoddir - Modify attributes of remote directories

You can use *ftmoddir* to modify the following attributes of a directory in a remote system:

– Directory name
– Access rights (not if the partner system is a Windows system or the partner is an FTP partner)

Format

```
ftmoddir -h |
          <partner 1..200>![<file name 1..512>]
        [ <transfer admission 8..67> | @n | @d |
          <user ID 1..67>[,[<account 1..64>][,[<password 1..64>]]] ]
        [ -p=[<management password 1..64>] ]
          -nf=<new file name 1..512> |  -am=@rw | -am=@ro
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

partner**!**[file name]
Specifies the directory and partner system for the attribute modification operation.

partner
partner is the name of the partner system in the partner list or the address of the partner system. For details on address specifications, see section “Defining the partner computer” on page 90.

file name
Name of the directory whose attributes are to be modified. The name can be either absolute or relative to the remote login admission. If the file name in the remote system has been predefined in an admission profile, it must not be specified here.

If the partner system is running openFT for BS2000/OSD then the name of a PLAM library can also be specified here.

transfer admission | **@n** | **@d** |
user ID[**,**[account][**,**[password]]]

Before you can modify the attributes of a file on a remote system, you must first identify yourself at the system. To do this, you need an authorization in the syntax used at the remote system. You can specify this transfer admission

– as an FTAC transfer admission if FTAC is used in the remote system,

– or as a login/LOGON admission in the syntax used by the remote system (user ID, possibly together with account or password).

For details, see section “Transfer admission” on page 94.

@n for *transfer admission*
By entering *@n* you specify that the remote system requires no login admission.

@d for *transfer admission*
If you specify *@d* (blanked) then the transfer admission is queried on the screen after the command has been sent. Your input is invisible to prevent unauthorized persons from seeing the transfer admission.

A binary password or binary transfer admission must be specified in hexadecimal format in the form x'...' or X'...'. If you enter the password directly, remember to invalidate the single quotes with a backslash (\) unless you have enclosed the remote login authorization in double quotes, for example X\'c6d9e4c5\'.

password not specified
If you omit a password which is required for authorization then it is queried on the screen after the command has been sent. Your input is invisible to prevent unauthorized persons from seeing the password. In this case, single quotes must not be invalidated with a backslash (\).

Please note that you still have to enter the commas, for example:

```
ftmoddir partner!file user-id,,
```
or
```
ftmoddir partner!file user-id,account,
```

neither *transfer admission* nor *user ID* specified

This has the same effect as *@d*, i.e. the transfer admission is queried on the screen after the command has been sent. openFT always interprets your (hidden) input as a transfer admission and not as a user ID.

-p=[management password]

If the directory is protected by a password in the remote system then you must specify this here.

The password must be specified in hexadecimal format in the form x'...' or X'...'. This is of relevance in the case of a connection with openFT for BS2000/OSD since it is possible to define hexadecimal passwords in BS2000. If you enter the password directly, remember to invalidate the single quotes with a backslash (\), for example: -p=X\'c6d9e4c5\'.

management password not specified

If you specify *-p=* then the password is queried on screen after the command has been sent. Your input is invisible to prevent unauthorized persons from seeing the password. In this case, single quotes must not be invalidated with a backslash (\).

-nf=new file name

Specifies the new name for the directory *file name* in the partner system. The name *file name* then loses its validity. *New file name* may be specified either absolutely or relative to the remote login admission.

-nf not specified

The directory name is unchanged.

-am=@rw | **-am=@ro**

Modifies the access rights to the directory *file name* in the remote system. If the partner system is a Windows system, you cannot change the access rights. For Unix or BS2000 systems you can specify either *@rw* or *@ro*:

@rw means that the access right is *read-write*.

@ro means that the access right is *read-only*.

-am not specified

No change is made to the access right definitions.

Examples

1. The directory *d:\dir* in the remote Windows system *win1* is to be renamed to *d:\users\\dir*, the transfer admission is *ChangeDirwin*:

```
ftmoddir win1!d:\\dir ChangeDirwin -nf=d:\\users\\dir
```

2. The directory */home/user1/current* in the remote Unix system *ux1* is to be renamed to */home/user1/previous*, the transfer admission is *Changedirux*:

```
ftmoddir ux1!/home/user1/current ChangeDirux  -
          -nf=/home/user1/previous
```

5.18 ftmodf - Modify the FTAM attributes of a local file

This command is above all useful in connection with FTAM partners.

For openFT partners, files of type *binary-fixed* can be provided (see also "Example for openFT partners" on page 230). The attributes *file type*, *record format* and *record length* are also evaluated when sending a file to openFT partners, but are not set when creating the receive file.

With *ftmodf*, you can modify the FTAM attributes of a file in the local system for a file transfer or file management request involving an FTAM partner. You can also delete the information in the FTAM catalog without deleting the file itself.

The following attributes can be defined:

– File type
– Character set
– Record format
– Record length
– FTAM access rights for a file that cannot be changed by the FTAM partner (permitted actions).

File attributes for file type, character set and record format may only be changed if you are aware of the file contents. If this is not the case, file inconsistencies occur, with the result that data transfer requests to the affected files are terminated. Consult the table that describes the operands.

Note that you cannot use *ftmodf* to negate file attributes on the Unix system. This means that a file can be deleted by means of operating-system resources (for example *rm*) even if the *permitted actions* do not permit deletion by an FTAM partner.

Format

```
ftmodf  -h |
          <file name 1..512>  -np=@d  |
          <file name 1..512>
        [ -ft=t | -ft=b ]
        [ -cs=g | -cs=c | -cs=i | -cs=v ]
        [ -rf=v | -rf=f | -rf=u ]
        [ -rl=<1..32767> ]
        [ -pa=[n][r][i][p][x][e][a][c][d] ]
        [ -np=<file access password 1..11> | -np=@n ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

file name **-np=@d**
Deletes all the information on the specified file in the FTAM catalog without deleting the file itself. *-np=@d* should not be specified together with other parameters, as these then have no effect.

file name
file name without *-np=@d* indicates the file in the local system whose attributes are to be modified. The file name can be either absolute or relative.

-ft=t | **-ft=b**
This identifies the type of file in the local system. You can enter either *t* or *b*.

t The file contains text data.

b The file contains binary data.

-ft not specified
The previous file type remains unchanged.

-cs=g | **-cs=c** | **-cs=i** | **-cs=v**
This can only be used in conjunction with the *t* (text) file type, and describes the character set for the text file, see also *universal class number* in section “FTAM files” on page 79. This attribute only has any point in the case of FTAM partners.

g GraphicString

The file can contain characters from the G0 set defined in ISO646 or ISO8859-1, or from the G1 set defined in ISO8859-1.

c GeneralString

The file can contain characters from the C0 set defined in ISO646, the G0 set defined in ISO646 or ISO8859-1, or the G1 set defined in ISO8859-1. In the case of transfer with FTAM partners, each set is terminated with a CRLF (Carriage Return Line Feed); in this case, set boundaries do not necessarily correspond to the transfer unit boundaries.

i IA5String

The file can contain characters from the C0 set and the G0 set defined in ISO646. In the case of transfer with FTAM partners, each set is terminated with a CRLF (Carriage Return Line Feed); in this case, set boundaries do not necessarily correspond to the transfer unit boundaries.

v VisibleString

The file can contain characters from the G0 set defined in ISO646.

-cs not specified
The previous character set remains unchanged.

-rf=v | **-rf=f** | **-rf=u**
This indicates how the data is to be transferred to an FTAM partner.

v (variable)
The data is transferred to an partner in records of variable length. Please note that, in the case of FTAM partners, in accordance with the A/111 profile, only text data from the GraphicString or VisibleString character sets can be transferred in this way. Binary files in a user format (where a record comprises a record length field and the data) can only be transferred to an FTAM partner in records of variable length, if the FTAM partner supports the userformat.

f (fix)
The data is transferred to an partner in records of equal length. Please note that, in the case of FTAM partners, in accordance with the A/111 profile, only text data from the GraphicString or VisibleString character sets can be transferred in this way.

Binary files of fixed record length (the file is made up of records of identical length) can only be transferred to an FTAM partner if the partner supports this fixed length for binary files.

u (undefined)
The record length used to transfer the data is not mapped to the real system. This means that the record length used for the transfer is not identical to that in the real file.

Binary files are stored in a bit string in the real system. Please note that in accordance with the A/111 profile, it is only possible to transfer text data from the GeneralString or IA5String character sets, or binary data with this record format. Any record structure

present in text files is also lost unless maintained using other mechanisms (e.g. CRLF line separation for the transfer of IA5 or GeneralString files with FTAM).

-rf not specified
: The previous record format remains unchanged.

-rl=record length
: Defines the record length in bytes with which the data is to be transferred to an FTAM partner. The maximum record length is 32767 bytes.

-rl not specified
: The previous record length remains unchanged.

-pa=[**n**][**r**][**i**][**p**][**x**][**e**][**a**][**c**][**d**]
: Defines the "permitted actions" and how an FTAM partner can access a local file. This parameter does not affect the access rights of a file in a Unix system but instead places additional constraints on the access possibilities for FTAM partners.

  The following values can be specified for the *permitted actions* parameter: *n*, *r*, *i*, *p*, *x*, *e*, *a*, *c*, *d*, or any combination of these values:

  n means that an FTAM partner cannot access this file. If *n* is specified, all other options are ignored.

  r means that an FTAM partner can read the file.

  r not specified
  : The file cannot be read.

  i with FTAM partners means that the FTAM partner can insert data units, such as records, in the file.

  i not specified
  : No data units can be inserted in the file.

  p means that an FTAM partner can overwrite the file.

  p not specified
  : The file cannot be overwritten.

  x means that an FTAM partner can append data to the file.

  x not specified
  : The file cannot be extended.

  e with FTAM partners means that the FTAM partner can delete data units, such as records, from the file.

e not specified
No data units can be deleted from the file.

a means that an FTAM partner can read the attributes of the file.

a not specified
The file attributes cannot be read.

c means that an FTAM partner can change the attributes of the file.

c not specified
The file attributes cannot be changed.

d means that an FTAM partner can delete the file.

d not specified
The file cannot be deleted.

-pa not specified
The access rights remain unchanged.

-np=file access password | **-np=@n**
This parameter is reserved for special customer applications.

For *file type*, *character set*, and *record format*, you should select combinations that correspond to the file contents:

Entries for	**-ft=**	**-cs=**	**-rf=**
Text files	t t t t t t	g g v v c i	f v f v u u
Structured binary files	b	No entry	v
Unstructured binary files	b	No entry	u
Binary files with fixed record length	b	No entry	f

Otherwise, file inconsistencies may occur. File access errors are also possible if the record format is set to *f*, but no record length is specified or the file size is not a multiple of the record length.

Example for FTAM partners

You wish to reset the access rights of the local file junk such that no FTAM partner can access the file.

```
ftmodf␣junk␣-pa=n
```

Example for openFT partners

The combination of *-ft=b* and *-rf=f* is also significant for file transfer with the openFT protocol. In this way, a BS2000 partner, for example, can fetch a file containing binary data from a Unix system and store it in BS2000 as a SAM file. To do this, the following entries are required in the Unix system and BS2000 systems.

Unix system:

```
ftmodf␣binfix06␣-ft=b␣-rf=f␣-rl=14156
```

BS2000:

```
ncopy␣from,ftunix,(binfix06,l=*n), -
     *a(´binfix.06´,,´binfixprofile´),data=*bin
```

5.19 ftmodp - Modify FT profiles

ftmodp stands for "modify profile".

You can use this command to modify your FT profiles. If an FT profile has been privileged, you can use *ftmodp* to remove its privileged status or change the transfer admission.

The timestamp is updated when a profile is modified.

Format

```
ftmodp -h |
          <profile name 1..8> | @s | @a
        [ -s=[<transfer admission 8..32> | @a | @n ]
              [,<user ID 1..32> | @a | @adm] ]
        [ -ua=[ <user ID 1..32> ],[<password 1..20> | @n ] ]
        [ -nn=<profile name 1..8> ]
        [ -tad= | -tad=<transfer admission 8..32> | -tad=@n ]
        [ -v=y | -v=n ]     [ -d=yyyymmdd | -d= ]
        [ -u=pr | -u=pu ]   [ -priv=y | -priv=n ]
        [ -iml=y | -iml=n ]
        [ -iis=y | -iis=n ]   [ -iir=y | -iir=n ]
        [ -iip=y | -iip=n ]   [ -iif=y | -iif=n ]
        [ -ff= | -ff=[t][m][p][r][a][l] | -ff=c ]
        [ -dir=f | -dir=t | -dir=ft ]
        [ -pn=<partner 1..200>,...,<partner(50) 1..200> | -pn= ]
        [ -pna=<partner 1..200>,...,<partner(50) 1..200> ]
        [ -pnr=<partner 1..200>,...,<partner(50) 1..200> ]
        [ -fn=<file name 1..512> | -fn=] [ -fnp=<file name prefix 1..511> ]
        [ -ls= | -ls=@n | -ls=<command1 1..1000> ]
        [ -lsp= | -lsp=[<command2 1..999> ][ -lss= | -lss=command3 1..999> ]
        [ -lf= | -lf=@n | -lf=<command4 1..1000> ]
        [ -lfp= | -lfp=<command5 1..999>][-lfs= | -lfs=<command6 1..999> ]
        [ -wm=o | -wm=n | -wm=e | -wm=one ]
        [ -c= | -c=y | -c=n ]
        [ -txt=<text 1..100> | -txt= ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

profile name
specifies the name of the FT profile you wish to modify. To see the profile names you have already assigned, you can issue the *ftshwp* command (without options).

@s for *profile name*
@s allows you to change the properties of the standard admission profile of the user ID.

The options *-v*, *-d* and *-u* are ignored with a standard admission profile.

@a for *profile name*
modifies all FT profiles that come into question at once, unless you select a specific profile with the option *-s*.

If you specify *ftmodp profile name* without any other parameters, you force the timestamp of the profile to be updated.

-s=[transfer admission | **@n** | **@a**][**,**user ID | **@a** | **@adm**]
is used to specify selection criteria for the FT profile to be modified.

transfer admission
specifies the transfer admission of the FT profile to be modified. You must specify a binary transfer admission in the form x\'...\' or X\'...\' .

@a for *transfer admission*
modifies either the FT profile specified with *profile name* (see above) or (if no profile name was specified) all the profiles that come into question.

If you specify *@a* as a user, you must specify a login name for *login name* (not *@a*). Otherwise, an error message is received.

@n for *transfer admission*
selects all FT profiles without transfer admission.

transfer admission not specified
causes to query the transfer admission on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the transfer admission. To

exclude the possibility of typing errors, the program prompts you to enter the transfer admission a second time. If you just press <ENTER>, this has the same effect as specifying *@a*.

,user ID
: As user, you can only enter your own login name here.

@a for *user ID*
: allows each user to modify only profiles belonging to his or her own login name. If *@a* is specified here, a transfer admission must be specified for *transfer admission* (not *@a*). Otherwise, an error message is received.

@adm for *user ID*
: For the FTAC and ADM administrator only.

user ID not specified
: modifies only profiles belonging to the user's own login name, regardless of who issues the command.

-s not specified
: if *@a* is specified for *profile name*, all the FT profiles belonging to the login name under which the *ftmodp* command is issued are modified. Otherwise, the FT profile with the specified name is modified.

-ua=[user ID]**,**[password | **@n**]
: *-ua* is only meaningful for the FTAC administrator in order to assign any desired FT profile of a login name to another login name.

user ID
: As user, you can only specify your own login name here.

,password
: specifies the password for a login name. A binary password must be specified in the form x\'...\' or X\'...\' . The FT profile for the login name is valid only so long as the password *password* is valid for the login name. When the password is changed, the profile can no longer be used (not locked!).

@n for *password*
: Can only be specified by the FTAC administrator!

comma only (,) no *password* specified
: causes FTAC to query the password on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the transfer admission. In this case, single quotes must not be escaped by a backslash (\).

user ID only (without comma and *password*) specified
means that the profile is valid again for all passwords of the specified login name *user ID*.

*-ua␣*not specified
the login name of this FT profile remains unchanged.

-nn=profile name | **@s**
-nn can be used to assigns a new name to one of your FT profiles.

@s for *profile name*
Makes the admission profile the standard admission profile for the user ID. If the admission profile previously had a transfer admission, you must also specify *-tad=@n*.

-nn not specified
leaves the profile name unchanged.

-tad=[transfer admission | **@n**]
allows you to modify the transfer admission of an FT profile. If the modified admission profile is a standard admission profile (*ftmodp @s* or *-nn=@s*), only *-tad=@n* is permitted.

transfer admission
The transfer admission must be unique within your Unix system so that there are no conflicts with transfer admissions defined by other FTAC users for other access permissions. A binary transfer admission must be specified in hexadecimal format in the form x\'...\' or X\'...\' . If the transfer admission you select has already been assigned, FTAC rejects the *ftmodp* command and issues the message
`Transfer admission already exists.`

@n for *transfer admission*
disables the old transfer admission.

@n must be specified if you convert an admission profile that has a transfer admission to a standard admission profile using *-nn=@s*.

transfer admission not specified
-tad= causes FTAC to prompt you to enter the transfer admission after the command has been entered. Your entry is not displayed to prevent unauthorized persons from seeing the transfer

admission. To exclude the possibility of typing errors, the program expects you to enter the transfer admission a second time as an entry check.

The transfer admission is not queried when a standard admission profile is changed. The following message is issued: `Transfer admission of standard profile must be @n.`

-tad not specified
: does not modify the transfer admission of the FT profile.

-v=y | **-v=n**
: *-v* defines the status of the transfer admission.

  y the transfer admission is not disabled (it is valid).

  n transfer admission is disabled (it is not valid).

  -v is ignored if the modified profile is a standard admission profile.

-v not specified
: the transfer admission status remains unchanged.

-d=[yyyymmdd]
: *-d* specifies the period during which the transfer admission can be used. The FT profile is disabled when this period has expired.
  You can specify an eight-digit date (e.g. 20170602 for June 2, 2017). The transfer admission can no longer be used after 0:00 hours on the specified day. The largest possible value that can be specified for the date is 20380119 (January 19, 2038).

  yyyymmdd not specified
  : when *-d=* is specified, the previous setting is cancelled, i.e. the time restriction is removed from the transfer admission.

  -d is ignored if the modified profile is a standard admission profile.

-d not specified
: the previous time restriction defined for the transfer admission remains unchanged.

-u=pr | **-u=pu**
: using *-u*, you can control how FTAC reacts when someone attempts to assign an existing transfer admission to an FT profile. Normally, the transfer admission must be disabled immediately, by designating it as private.

Transfer admissions that do not require as much protection, can be designated as public. This means that they are not disabled even when a user attempts to assign another transfer admission of the same name. Possible values:

pr (default value)
the transfer admission is disabled as soon as someone with another login name attempts to specify a transfer admission of the same name (private).
In this case, the *-u* and *-d* parameters are set to their default values at the same time.

pu the transfer admission is not disabled, even if someone attempts to specify a transfer admission of the same name (public).

-u is ignored if the modified profile is a standard admission profile.

-u not specified
the previous setting remains unchanged.

-priv=y | **-priv=n**
As a normal FTAC user, you can only withdraw an existing privilege. *y* is not permitted.

n withdraws the privileged status, if it had been granted, from the FT profile.

-priv not specified
does not modify the privileged status of the FT profile.

-iml=y | **-iml=n**
-iml (ignore max. level) is used to specify whether the FT profile is to be restricted by the values in the admission set. The user can override the entries he/she made himself or herself (the MAX. USER LEVELS) for requests using this FT profile. If the FT profile is also privileged by the FTAC administrator, the entries made by the FTAC administrator (the MAX. ADM LEVELS) can also be ignored. This FT profile would then allow *inbound* basic functions to be used which are disabled in the admission set.

y allows the values in the admission set to be ignored.

n restricts the functionality of the profile to the values in the admission set.

-iml not specified
: causes the values specified in the profile for the basic functions to apply unchanged.

-iis=y | **-iis=n**
: *-iis* (ignore inbound send) allows the value for the basic function *inbound send* in the admission set to be ignored (for details, see *-iml*).

  y allows the basic function *inbound send* to be used even if it is disabled in the admission set. At the same time, component "display file attributes" of the basic function *inbound file management* can be used (see table at *-iif*).

  Specifying this option is enough as long as the basic function *inbound send* was disabled by the user, but if it was disabled by the FTAC administrator, it is also necessary that he/she grant privileged status to the FT profile.

  n restricts the profile to the value in the admission set for the basic function *inbound send*.

-iis not specified
: causes the values specified in the profile for the basic function *inbound send* to apply unchanged.

-iir=y | **-iir=n**
: *-iir* (ignore inbound receive) allows the value for the basic function *inbound receive* in the admission set to be ignored (for details, see *-iml*).

  y allows the basic function *inbound receive* to be used even if it is disabled in the admission set. At the same time, subcomponents of the basic function *inbound file management* can also be used (see table at *-iif*).

  Specifying this option is enough as long as the basic function *inbound receive* was disabled by the user, but if it was disabled by the FTAC administrator, it is also necessary that he/she grant privileged status to the FT profile.

  n restricts the profile to the value in the admission set for the basic function *inbound receive*.

-iir not specified
: causes the values specified in the profile for the basic function *inbound receive* to apply unchanged.

-iip=y | **-iip=n**

-iip (ignore inbound processing) allows the value for the basic function *inbound follow-up processing + preprocessing + postprocessing* in the admission set to be ignored (for details, see *-iml*).

y allows the basic function *inbound follow-up processing + preprocessing + postprocessing* to be used even if it is disabled in the admission set. Specifying this option is enough as long as the function was disabled by the user, but if it was disabled by the FTAC administrator, it is also necessary that he/she grant privileged status to the FT profile.

n restricts the profile to the value in the admission set for the basic function *inbound follow-up processing + preprocessing + postprocessing*.

-iip not specified

causes the values specified in the profile for the basic function *inbound follow-up processing + preprocessing + postprocessing* to apply unchanged.

-iif=y | **-iif=n**

-iif (ignore inbound file management) allows the values for the basic function *inbound file management* in the admission set to be ignored (for details, see *-iml*).

y allows the basic function *inbound file management* to be used even if it is disabled in the admission set.

Specifying this option is enough as long as the basic function *inbound file management* was disabled by the user, but if it was disabled by the FTAC administrator, it is also necessary that he/she grant privileged status to the FT profile.

n restricts the profile to the value in the admission set for the basic function *inbound file management*.

The following table shows which subcomponents of the file management can be used under which conditions.

Inbound file management function	**Values of the admission set or extension in profile**
Display file attributes	Inbound Send (IBS) enabled
Modify file attributes	Inbound Receive (IBR) **and** Inbound File Management (IBF) enabled

Inbound file management function	**Values of the admission set or extension in profile**
Rename files	Inbound Receive (IBR) **and** Inbound File Management (IBF) enabled
Delete files	Inbound Receive (IBR) enabled **and** Write mode = overwrite in profile
Display directories	Inbound File Management (IBF) enabled
Create, rename and delete directories	Inbound File Management (IBF) enabled **and** direction = from partner in profile

-iif not specified
causes the values specified in the profile for the basic function *inbound file management* to apply unchanged.

-ff=[**t**][**m**][**p**][**r**][**a**][**l**] | **-ff=c**
-ff defines the FT function for which the FT profile can be used. With the exception of *c*, these letters can be combined in any way (*tm, mt, mr,* ...). *c* must not be combined with other values. Please observe the note concerning the description of *-ff=c* on page 240.

t (transfer) The FT profile can be used for the file transfer functions “Transfer files”, “Display file attributes”, and “Delete files”.

m (modify file attributes) The FT profile can be used for the file transfer functions “Display file attributes” and “Modify file attributes”.

p (processing) The FT profile can be used for the file transfer functions "File Preprocessing" or "File Postprocessing". The FT function „Transfer files“ must also be permitted.

Specification of *p* has no significance for profiles with a file name prefix (*-fnp=*) or a file name (*-fn=*) since, in this case, the first character of the file name or file name prefix decides whether the profile can only be used for preprocessing and postprocessing ("|") or only for file transfer/file management (no "|").
The use of follow-up processing is not controlled by *-ff=*, but by *-lf=* and *-ls=*.

r (read directory) The FT profile can be used for the file transfer functions “Display directories” and “Display file attributes”.

a (administration)
The admission profile is allowed to be used for the "remote administration" function.
-ff=a may only be specified by the FT administrator or FTAC administrator.

l (logging)
The admission profile is allowed to be used for the "Receive ADM traps" function.
-ff=l may only be specified by the FT administrator.

c (client access)
The admission profile is allowed to be used for the "access to remote administration server" function (ADM profile). *ff=c* may only be specified by the ADM administrator.

The value *c* must not be combined with any other value. In addition, an FT profile created with *-ff=c* cannot be changed into a FT profile using the other FT functions (*t, m, p, r, a* or *l*) and vice versa.

No function specified
Specifying *-ff=* allows you to undo any specification with regard to the functions. All file transfer functions are then permitted (corresponds to *tmpr*), but not the remote administration functions (*a, c*) and ADM trap functions (*l*).

-ff not specified
The previous specification with respect to the functions remains unchanged.

-dir=f | **-dir=t** | **-dir=ft**
specifies for which transfer direction(s) the FT profile may be used. Possible values for the direction: *f*, *t*, *ft*, *tf*.

f allows data transfer only from a partner system to the local system.

t allows data transfer only from the local system to the remote system. It is thus not possible to create, rename or delete directories.

ft, tf
transfer direction is not restricted in the profile.

-dir not specified
leaves the transfer direction entries in the FT profile unchanged.

-pn=[partner1[**,**partner2, ...]]

You use *-pn* to specify that this admission profile is to be used only for FT requests which are processed by a certain partner system. You can specify the name of the partner system in the partner list or the address of the partner system. For details on address specifications, see section “Defining the partner computer” on page 90.

You can specify more than one partner system (maximum 50) with a maximum total of 1000 characters.

partner1[,*partner2, ...*] not specified

-pn= cancels a previous restriction defined for partner systems so that the FT profile can be used by every partner system.

-pna=partner1[**,**partner2, ...]

-pna adds one or more partner system(s) to the list of permitted partner systems. Up to 50 partner systems can be entered in the list (max. 1000 characters).

If the list has been empty up to now, then the profile is limited to the specified partner system(s).

-pnr=partner1[**,**partner2, ...]

-pnr deletes one or more partner system(s) from the list of permitted partner systems.

Please note: As soon as you delete the last partner remaining in the list, the profile can be used by every partner system.

-pn, *-pna* and *-pnr* not specified

causes the entries for permitted partner systems to apply unchanged.

-fn=[file name]

-fn specifies which file(s) under your login name may be accessed using this FT profile. If you specify a fully qualified file name, only the file with this name can be transferred.
If the file name ends with %unique or %UNIQUE, this string is replaced by a string which changes for each new call. In Unix systems, this string is 14 characters long. In addition, a suffix separated by a dot may be specified after %unique or %UNIQUE, e.g. `file1%unique.txt`. Only the already converted file name is displayed in both the log and the messages.
If *file name* starts with a "|" (pipe character) then it is interpreted as a preprocessing or postprocessing command, see also section “Preprocessing and postprocessing” on page 100.

file name not specified

-fn= allows you to cancel a file name entry. This also applies to a prefix assigned with *-fnp*. The FT profile then permits unrestricted access to all files.

-fn not specified

leaves the file name entries in the FT profile unchanged.

-fnp=file name prefix

restricts access to a set of files whose names begin with the same prefix. FTAC adds the character string specified as *file name prefix* to the file name in the request and attempts to transfer the file with the expanded name.
For example, if this option is specified as *-fnp=scrooge/* and the request contains the file name *stock*, the file is transferred as *scrooge/stock*.

In this way, you can designate the files you have released for transfer. If the *-fnp* option was used to specify a prefix, the file name specified in the request must not contain the character string *../* to avoid (unintentionally) changing directories. You should also ensure that there is no chance for a symbolic link to cause a jump to another place in the file tree.
%unique or %UNIQUE cannot be used for a file name prefix. In the case of a file transfer request, the user can use a file name ending with %UNIQUE (or %UNIQUE.*suffix* or %unique or %unique.*suffix*) to generate a unique file name with the prefix specified here.

A file name prefix which starts with the | character indicates that the FTAC profile can only be used for file transfer with preprocessing and postprocessing, since the file name created using the prefix and the name specified for the ncopy or *ft* command also starts with the | character. In this case, no follow-up commands may be specified.

On Unix systems, the shell metacharacters | ; & < > and "newline" may only be specified if they are enclosed in '...' (single quotes) or "..." (double quotes) or if each of them is escaped with "\" (backslash). The character ` (accent grave) and the string $((dollar+open bracket) may only be specified if they are enclosed in '...' (single quotes) or if they are specified directly after a backslash ("\").

The following strings may not be specified:
– .. (two dots)
– .\ (dot + backslash)
– .' (dot + single quote)
This makes it impossible to navigate to higher-level directories.

file name prefix can be up to 511 characters in length.

-fn= allows you to cancel a file name prefix entry, see above.

Special cases

- You must specify a file name or file name prefix which starts with the string "|ftexecsv␣" for FTAC profiles which are to be used exclusively for the *ftexec* command. If a command prefix is also to be defined, you must specify it as follows:

  -fnp="|ftexecsv␣-p=*command prefix***"**
  (e.g.: `-fnp="|ftexecsv␣-p=\"ftshwr␣\""`)

  The same restrictions apply to the command string of the *ftexec* call as to the filename prefix during preprocessing and postprocessing.

- For FTAC profiles that are only to be used for getting monitoring data, specify the filename prefix "|*FTMONITOR ". The functions of the profile must permit File Preprocessing (*-ff=tp*). For details, see the *ftcrep* command, Example 3 on page 189.

-fnp not specified
leaves the *file name prefix* entries in the FT profile unchanged.

-ls= | **-ls=@n** | **-ls=**command1
specifies follow-up processing which is to be performed under your login name in the event that **file transfer** is **successful**. If *-ls* is specified, no success follow-up processing may be requested in the FT request. Specifying *-ls* only makes sense if you also make an entry for *-lf* (see below) to preclude the possibility that an intentionally unsuccessful request can circumvent the *-ls* entry. If you have defined a prefix for the file name with *-fnp* and plan follow-up processing for this file, you must specify the complete file name here.

@n for *command1*
If you enter *-ls=@n*, no follow-up processing is then permitted in the FT profile in the event that file transfer is successful.

command1 not specified
-ls= allows you to cancel a follow-up-processing entry. The FT profile then no longer restricts success follow-up processing in the local system. This is also a way to cancel a prefix for the follow-up processing defined with *-lsp*.

-ls not specified
leaves the entries in the FT profile for follow-up processing in the event that file transfer is successful unchanged.

-lsp=[command2]
-lsp defines a prefix for follow-up processing in the local system in the event that **file transfer** is **successful**. FTAC then adds the character string *command2* to the follow-up processing specified in the FT request and attempts to execute the resulting command. For example, if this option is specified as *-lsp='lpr␣'* and the request specifies *file-name* as follow-up processing, FTAC executes *lpr␣file-name* as follow-up processing.
Prefix, suffix and follow-up processing commands must together not be longer than 1000 characters.

Please also bear in mind the information provided on the *-ls* option!

If a prefix was defined with *-lsp*, the character set available for specifying follow-up processing in the FT request is restricted to:
– alphanumeric characters (letters and digits)
– the special characters + = / ! ␣ - , @ _ " $ '
– a period (.) between alphanumeric characters

You can cancel an existing prefix by specifying *-ls=*.

command2 not specified
-lsp= cancels the entry in the FT profile for a follow-up processing prefix after successful file transfer.

-lsp not specified
leaves the prefix entries in the FT profiles for follow-up processing in the event that file transfer is successful unchanged.

-lss=[command3]
-lss defines a suffix for follow-up processing in the local system in the event that **file transfer** is **successful**. FTAC then appends the character string *command3* to the follow-up processing specified in the FT request and attempts to execute the resulting command. For example, if this option is specified as *-lss=␣file-name* and the request specifies *lpr* as follow-up processing, FTAC executes *lpr␣file-name* as follow-up processing.
Prefix, suffix and follow-up processing commands must together not be longer than 1000 characters.
Please also bear in mind the information provided on the *-ls* option!

If a suffix was defined with *-lss*, the character set available for specifying follow-up processing in the FT request is restricted to:
- alphanumeric characters (letters and digits)
- the special characters + = / ! ␣ - , @ _ " $ ’
- a period (.) between alphanumeric characters

command3 not specified
: *-lss*= cancels the entry in the FT profile for a follow-up processing suffix after successful file transfer.

-lss not specified
: leaves the suffix entry in the FT profile for follow-up processing unchanged.

-lf= | **-lf=@n** | **-lf=**command4
: *-lf* specifies follow-up processing to be executed under your login name if the **file transfer** is **aborted** due to an error. If *-lf* is specified, no failure follow-up processing may be requested in the FT request. Making an *-lf* entry only makes sense if you also make an entry for *-ls* (see above) to preclude the possibility that a successful request can circumvent the *-lf* entry. If you have defined a prefix for the file name with *-fnp* and plan follow-up processing for this file, you must specify the complete file name here.

@n for *command4*
: *-lf=@n* is specified, no follow-up processing is then permitted in the FT profile in the event of an unsuccessful file transfer.

command4 not specified (*-lf=*)
: *-lf=* allows you to cancel an entry for follow-up-processing in the event that file transfer is unsuccessful. The FT profile then no longer restricts failure follow-up processing in the local system. This is also a way to cancel a prefix defined with *-lfp*.

-lf not specified
: leaves the entries in the FT profiles for failure follow-up processing after unsuccessful file transfer unchanged.

-lfp=[command5]
: defines a prefix for follow-up processing in the local system in the event that **file transfer** is **unsuccessful**. FTAC then adds the character string *command5* to the follow-up processing specified in the FT request and attempts to execute the resulting command. For example, if this option is specified as *-lfp=’lpr␣’* and the request specifies *error.txt* as follow-up processing, FTAC executes *lpr␣error.txt* as follow-up processing.

Prefix, suffix and follow-up processing commands must together not be longer than 1000 characters.

Please also bear in mind the information provided on the *-lf* option!

If a prefix was defined with *-lfp*, the character set available for specifying follow-up processing in the FT request is restricted to:
– alphanumeric characters (letters and digits)
– the special characters + = / ! ␣ - , @ _ " $ ’
– a period (.) between alphanumeric characters

You can cancel an existing prefix by specifying *-lf*=.

command5 not specified
-lfp= cancels the follow-up processing prefix in the FT profile in the event of unsuccessful file transfer.

-lfp not specified
leaves the prefix entries in the FT profiles for follow-up processing in the event of unsuccessful file transfer unchanged.

-lfs=[command6]
-lfs defines a suffix for follow-up processing in the local system in the event that **file transfer** is **unsuccessful**. FTAC then appends the character string *command6* to the follow-up processing specified in the FT request and attempts to execute the resulting command. For example, if this option is specified as *-lfs=␣file-name* and the request specifies *lpr* as follow-up processing, FTAC executes *lpr␣file-name* as follow-up processing.
Prefix, suffix and follow-up processing commands must together not be longer than 1000 characters.

Please also bear in mind the information provided on the *-lf* option!

If a suffix was defined with *-lfs*, the character set available for specifying follow-up processing in the FT request is restricted to:
– alphanumeric characters (letters and digits)
– the special characters + = / ! ␣ - , @ _ " $ ’
– a period (.) between alphanumeric characters

command6 not specified
-lfs= cancels the follow-up processing suffix in the FT profile in the event of unsuccessful file transfer.

-lfs not specified
leaves the suffix entry in the FT profile for a follow-up processing in the event of unsuccessful file transfer unchanged.

-wm=o | **-wm=n** | **-wm=e** | **-wm=one**

-wm specifies which write modes may be used in the file transfer request and what they effect.

o (overwrite) In the FT request of openFT or FTAM partners, only *-o* or *-e* may be entered for write mode. The receive file is overwritten if it already exists, and is created if it does not yet exist.

With FTP partners, *-n* may also be entered if the file does not yet exist.

n (no overwrite) In the FT request *-o*, *-n* or *-e* may be entered for write mode.
The receive file is created if it does not yet exist. If the receive file already exists, the request is not executed.

e (extend) In the FT request only *-e* may be entered for write mode, i.e. the receive file is extended by appending the transferred file to the end if the receive already exists. The receive file is created if it does not yet exist.

one means that the FT profile does not restrict the write mode.

-wm not specified

leaves the write-mode entries in the FT profile unchanged.

-c= | **-c=y** | **-c=n**

Using *-c*, you can determine whether data encryption is required or forbidden. If the setting in the profile does not correspond to the setting in the request, the request is denied. The setting is not valid for file management requests, since there is no encryption for these requests.

y Only requests **with** data encryption may be processed using this profile.

n Only requests **without** data encryption may be processed using this profile.

neither *y* nor *n* specified

-c= resets the current setting. Requests with and without data encryption are both accepted.

-c not specified

The encryption option remains unchanged.

-txt=[text]

-txt allows you to enter a new comment in the FT profile (up to 100 characters).

text not specified

-txt= deletes an existing comment.

-txt not specified

an existing comment remains unchanged.

As soon as you modify an admission profile, the timestamp is also updated. The timestamp is output with *ftshwp -l* (LAST-MODIF). The timestamp is also updated if you do not change the properties of the profile, i.e. if you enter *ftmodp* without any parameters.

CAUTION

If you use the *-ff=p*, *-fn*, *-fnp*, *-ls*, *-lsp*, *-lss*, *-lf*, *-lfp* or *-lfs* options, you must remember

- that a file name restriction can be bypassed by renaming the file unless follow-up processing is also restricted;
- that follow-up processing must always be restricted for both successful and unsuccessful file transfer and, if necessary, equivalent restrictions must exist for any permitted preprocessing;
- that prefixes for the file names and follow-up processing must be matched to one another;
- that no symbolic links should occur in the part of your file tree that is referenced by the file name prefix;
- that restrictions applied to preprocessing or follow-up processing can be circumvented if it is possible to replace this command with, for example, a "Trojan horse".

Example

The transfer admission in the *goldmrep* FT profile created in the “Example” on page 188, is to be changed to *forScrooge*. The transfer direction is no longer to be restricted. The profile is to be used to transfer any files with the prefix *mine/*. Follow-up processing is to be prohibited entirely.

The following command has to be entered:

```
ftmodp␣goldmrep␣-tad=forScrooge␣-dir=tf\
␣-fnp=mine/␣-ls=@n␣-lf=@n
```

5.20 ftmodr - Change the property of requests

With the *ftmodr* command, you can change the priority of requests you have issued, or of a group of requests, for example all the requests to a particular partner. Furthermore, you have the option of changing the order of requests within a priority.

Format

ftmodr -h |
[-ua=<user ID 1..32> | -ua=@a]
[-pn=<partner 1..200>]
[-fn=<file name 1..512>]
[-pr=n | -pr=l][-qp=f | -qp=l]
[<request ID 1..2147483647>]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-ua=user ID | **-ua=@a**
You use *-ua* to specify the user ID for which requests are to be modified. As a user, you can omit this specification since you may only enter your own user ID.

-ua= not specified
Your own user ID is the selection criterion.

-pn=partner
You use *-pn* to specify a name or an address for the partner system for which you want to modify requests. The partner should be specified in the same way as in the request or as it is output in the *ftshwr* command without the option *-s*, *-l* or *-csv*. If openFT finds a partner in the partner list that corresponds to the specified partner address then *ftshwr* indicates the name of the partner even if a partner address was specified on request entry.

-fn=file name
You use *-fn* to specify the file name for which requests are to be modified. Requests which access this file in the local system are modified.

You must specify the file name that was used when the request was created. This file name is also output by the *ftshwr* command without the *-fn* option.

Wildcards may not be used in the file name.

-pr=n | **-pr=l**
indicates the new priority. The following values are possible:

n (normal)
the request has the priority "normal".

l (low)
the request has the priority "low".

-qp=f | **-qp=l**
indicates the position of the request within the same priority. The following values are possible:

f (first)
the request is placed at the top of the list of requests with the same priority.

l (last)
the request is placed at the bottom of the list of requests with the same priority.

request ID
request ID is used to specify the identification of a specific request that is to be modified. The request ID is output on the screen when reception of the request is confirmed. It can also be displayed using the *ftshwr* command.

If you have specified a request ID but the other specified selection criteria do not match the request then the request is not modified and the following error message is output:

```
ftmodr: Request request ID not found
```

5.21 ftmonitor - Call the openFT Monitor for displaying measurement data

The *ftmonitor* command calls the openFT Monitor in which the monitoring data collected during openFT operation is displayed. openFT can be running on the local system or on a remote system. The openFT Monitor can only be called if monitoring has been explicitly activated by the administrator on the relevant system and the asynchronous openFT has been started.

Note that you require a graphics-capable terminal to use the *ftmonitor* command.

Format

ftmonitor -h |
[-lay=<monitor layout file name 1..512>]
[-po=<polling intervall 1..600>]
[<partner 1..200> [
<transfer admission 8..67> |
<user ID 1..67>[,[<account 1..64>][,[<password 1..64>]]]]]

Description

-h Outputs the command syntax. Any specifications after *-h* are ignored.

-lay=monitor layout file name
Name of the Monitor layout file. This file describes what monitoring data is output and how it is presented.

The name of the layout file must be specified with the suffix *.ftmc*. This suffix is automatically assigned by the monitor when the file is saved if it was not explicitly specified there.

The content of the layout file is also generated by the Monitor. You must not change the content of the layout file.

After the default Monitor window has been opened for the first time (without specifying *-lay*), you can create and save your own layout file. To do this, choose a different layout from the *View* menu of the Monitor window, for instance, or set a different value using the selection icon on the top right and store the setting under a name of your choice. Refer to the online Help system of the Monitor window for details.

-lay not specified

If you do not specify *-lay*, the default Monitor window is opened. This contains a chart showing the monitoring value *Networkb/sec of all Requests* (corresponds to the parameter *ThNetbTtl* in the command *ftshwm*).

-po=polling interval

Polling interval in seconds.

Possible values: 1 through 600.
Default value: 1

partner

Name or address of the partner system for which monitoring data is to be shown. The partner must be an openFT partner (i.e. communication via the openFT protocol) and must support the collection of monitoring data, i.e. the openFT version of the partner must be at least V11.

In addition, the partner's asynchronous openFT server must be started and monitoring must be activated in its operating parameters.

partner not specified

If you do not specify a partner, the monitoring data of the openFT instance on the local computer is output.

transfer admission | user ID[**,**[account][**,**[password]]]

Transfer admission for the partner system. File transfer and preprocessing/postprocessing must be permitted under the specified transfer admission.

You can specify this transfer admission

– as an FTAC transfer admission if FTAC is used in the remote system or destination instance. For this purpose, a special admission profile with the filename prefix *FTMONITOR can be set up on the partner system that only permits monitoring data to be collected. You will find an example under *ftcrep* on page 189.

– or as a login/LOGON admission using the syntax of the remote system (*user ID*, where necessary with *account* and/or *password*).

transfer admission not specified

If you do not specify a transfer admission for a remote partner system, the system prompts you for it in a dialog box. The entry made for the password or the FTAC transfer admission remains invisible. Asterisks (*****) are displayed as replacement characters.

Messages from the openFT Monitor

The openFT Monitor issues error messages in the form of a dialog box. It terminates automatically if an error occurs or if monitoring is terminated in the system being monitored.

If the layout of the Monitor window is changed and if openFT is terminated before the changed layout is saved, the openFT Monitor issues a message and queries whether the layout is to be saved.

5.22 ftmsg - Output a message box on a graphical display

The command *ftmsg* allows a message box to be output on the display defined by the DISPLAY variable.

ftmsg can be used to output messages on a graphical display from within local follow-up processing.

Please note that you require a graphics-capable terminal in order to use the *ftmsg* command.

Format

ftmsg [<window title>:]<message text>

Description

window title
: Title of the message box.
Default value for the title is "openFT".

message text
: Message text for the message box.

5.23 ftseti - Set an instance

The . *ftseti* command allows you to select the openFT instance with which you want to work.

Format

.␣ftseti -h | <instance 1..8>

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

instance
Name of the instance to be selected.

The command sets the OPENFTINSTANCE environment variable to the instance name.

It must be called with

```
. ftseti
```

so that OPENFTINSTANCE is set in the current shell. The *std* instance is set by default.

The first *ftseti* call sets an alias (*ftseti*=. *ftseti*) in the current shell that allows the preceding period to be dispensed with in subsequent calls.

In some variants of the Bourne shell, the transfer parameters are not forwarded when "." is used in a call.
It may therefore be necessary with a call from a Bourne shell (e.g. under *su*) to switch to the K shell (*ksh*).
Alternatively, the OPENFTINSTANCE environment variable can also be set manually or in scripts to the desired instance name and exported.

Messages of the ftseti command

If *ftseti* could not be executed properly, a self-explaining message is output. The exit code is not equal zero in this case.

5.24 ftshw - Display the attributes of one or more remote files

With *ftshw* you can display the attributes of a file or files in a directory in the remote system.

There are three options for displaying the attributes:

– List the names of the files in a directory
– Display a default selection
– Display all attributes of a file or of files in a directory, as requested from the partner system

A precise description of default output and detailed output can be found in the section “Description of file attribute display” on page 260.

Output is written to standard output.

Format

```
ftshw   -h |
        [ -d ]
          <partner 1..200>![<file name 1..512>]
        [ <transfer admission 8..67> | @n | @d |
          <user ID 1..67>[,[<account 1..64>][,[<password 1..64>]]] ]
        [ -p=[<management password 1..64>] ]
        [ -s | -l ][ -csv ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-d Specifies that the attributes of the files in a remote directory are to be displayed.

-d not specified
The attributes of the file *file name* specified in the command are displayed.

partner**!**[file name]
specifies the system and the file(s) of which the attributes have to be displayed.

partner

partner is the name of the partner system in the partner list or the address of the partner system. For details on address specifications, see section “Defining the partner computer” on page 90.

file name

file name can be either absolute or relative to the remote login admission. If the file name in the remote system has been predefined in an FTAC authorization profile, it must not be specified here.

If the *-d* option is specified, file name indicates a directory in the remote system.

If the partner system is running openFT for BS2000/OSD, elements from PLAM libraries may also be specified here (Syntax: Libname/Element type/Element name).

If openFT for z/OS is running on the partner system, members from PO libraries can also be output here (syntax: library name/library member).

transfer admission | **@n** | **@d** |
user ID [**,**[account][**,**[password]]]

To enable you to execute file management requests in the remote system, you must furnish the remote system with proof of identity. For this purpose, you will need login admission in the syntax valid for the remote system. You can specify this transfer admission

– as an FTAC transfer admission if FTAC is used in the remote system,

– or as a login/LOGON admission in the syntax used by the remote system (user ID, possibly together with account or password).

For details, see section “Transfer admission” on page 94.

@n for *transfer admission*

By entering *@n* you specify that the remote system requires no login admission.

@d for *transfer admission*

Specifying *@d* (blanked transfer admission) causes openFT to query the transfer admission on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the transfer admission.

A binary password and binary transfer admission must be specified in hexadecimal format in the form x'...' or X'...'. If you enter the password directly, remember to insert a backslash (\) to escape the single quotes if you did not enclose the remote login admission in double quotes, for example: X\'c6d9e4c5\'.

password not specified
: Omitting the password necessary for admission causes openFT to query the password on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the password. In this case, single quotes must not be escaped by a backslash (\).

  Nevertheless, you have to specify the commas, e.g.:

```
ftshw partner!file user-id,,
```
or
```
ftshw partner!file user-id,account,
```

neither *transfer admission* nor *user ID* specified
: causes the same as *@d*, i.e. openFT queries the transfer admission on the screen after the command is entered. Your (blanked) entry is always interpreted as transfer admission and not as user ID.

-p=[management password]
: If the file in the remote system is protected by a password, you must enter this password here.

  A binary password must be entered in hexadecimal form x'...' or X'...'. This is of relevance for links to openFT for BS2000/OSD, because BS2000 supports the definition of hexadecimal passwords. If you enter the password directly, remember to insert a backslash (\) to escape the single quotes, for example: X\'c6d9e4c5\'.

  management password not specified
  : Specifying *-p=* causes openFT to query the password on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the password. In this case, single quotes must not be escaped by a backslash (\).

-s Only the file name or the names of the files in the directory or the file name are output (short).

-l All information available on the remote file in the partner system is requested. However, only attribute values returned by the partner system can be displayed (long).

neither *-s* nor *-l* specified:

A standard scope of information should be displayed.
A precise description of standard output and of detailed output can be found in the following section.

-csv Specifying *-csv* indicates that the attributes of files on remote systems are to be output in the CSV format. The values in the output are separated by semicolons. If you specify *-csv*, output is always in the long form (analogous to *-l*) regardless of whether you also specify *-l* or *-s*.

-csv not specified

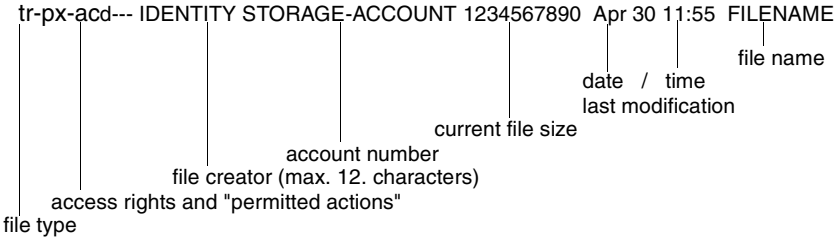
The attributes of files on remote systems are output in the standard format.

5.24.1 Description of file attribute display

The following section describes the output of the commands used to show the attributes of files on the local and remote systems. Both standard output and detailed output are described. The individual fields, their possible values and their meanings are listed.

The standard output is obtained if you do not specify the scope of the output; the detailed output is obtained only with a corresponding specification (see the following examples).

Standard output



Not all information is provided with the FTP protocol. Such missing information is replaced by '-' or by default values.

Detailed output, examples

```
$ ftshw bs2partn!aaa.e42 transbs2 -l

   FILENAME=:6QCA:$HUGO.AAA.E42
   CRE   HUGO DATE=Mar 17 13:01
   MOD   DATE=Mar 17 13:01
   REA   DATE=Mar 17 13:01
   BINARY-FILE
   RECORD-FORMAT=u RECORD-SIZE=32767
   ACCESS-RIGHTS=r-pxeacd---   FILESIZE=32768

$ ftshw zospart!test.clist transzos -l

   FILENAME=test.clist
   CRE   OPFTWIT
   MOD   DATE=Apr 03 2009
   RECORD-FORMAT=v RECORD-SIZE=648        FILE-AVAILABILITY=i
   ACCESS-RIGHTS=r-pxeacd---   FILESIZE=587860
```

Description of fields

file type

specifies the file type. This field can be assigned any of the following values:

t	File contains text
b	File contains binary data
d	Directory
*	No information available on the file structure

The comprehensive output is displayed as follows:

BINARY-FILE	Binary file
DIRECTORY	Directory
CHARACTERSET	Text file

The character set from which the characters in the text file originate is also specified for text files (CHARACTERSET=). The field can be assigned the following values:

g	GraphicString: the file can contain characters from the G0 set of ISO646, or from the G0 set of ISO8859-1 and the G1 set of ISO8859-1.
c	GeneralString: the file can contain characters from the C0 set of ISO646 and either from the G0 set of ISO646 or from ISO8859-1 and from the G1 set of ISO8859-1.
i	IA5String: the file can contain characters from the C0 set and the G0 set of ISO646.
v	VisibleString: the file can contain characters from the G0 set of ISO646.

access rights and permitted actions

contains information on the access rights which can be used for the file or the directory.

For files, this field can be assigned any of the following values:

r	File can be sent.
i	Units of data can be added.[1]
p	File can be overwritten.
x	File can be extended, i.e., data can be appended to it.
e	Units of data can be deleted from the file.
a	File attributes can be read.
c	File attributes can be modified.
d	File can be deleted.
t	Traversal [1]
v	Reverse traversal [1]
r	Random access [1]

[1]These values are only relevant for FTAM.

For directories (*-d* is specified), this field can be assigned any of the following values:

r	All files of the directory can be listed.
pxe	Under the directory, files and directories can be created, extended, and deleted.
a	Directory attributes can be read.
c	Directory attributes can be modified.
d	The directory can be deleted.

file creator
: identifies the creator of the file. In BS2000, the information refers to the user ID under which the file is created. In the Unix system, this value also identifies the owner of the file.
The field can be up to 12 characters in length.

STORAGE-ACCOUNT
: contains the account number used when calculating the cost of storing the file in the remote system.

If the partner returns an account number under FTAM, this is appended to the file owner in the standard output.

FILESIZE - current file size in bytes
: contains the current file size in bytes. If the output is followed by a "K", the output is in kilobytes. If it is followed by an "M", the output is in megabytes. This value is only as precise as the value returned by the partner system. Since files are created differently in different systems, different values can be displayed for files of the same size from different systems. Some filestores assign a multiple of a basic unit, e.g. blocks, for file storage. It is therefore advisable not to take this value to be the actual file size; it should be used for guidance only.

date and time of last modification to file contents
: contains information on when the file contents were last modified. In the case of modifications made within the last six months, the value is given in the form *month day time* (e.g. Jan 31 15:13); for earlier modifications, the form is *month day year* (e.g. Jan 31 2009).

FILENAME
: contains the name of the file.

The following values are part of the comprehensive output:

CRE, MOD, REA, ATM - how the file was last used
: contains information on how the file was last accessed. The following types of access are displayed:

CRE	Creating the file
MOD	Modifying the file contents (overwrite, extend)
REA *)	Reading the file (send)
ATM *)	Modifying the file attributes

*) These values are only relevant for FTAM.

It is important to remember that it is up to the remote system to determined which information it returns. Therefore, the information line on file use may look different and may contain different information, depending on the partner system. Generally, this section will at least indicate how the file was created.

However, additional information on modifying the file contents or file attributes, or sending a file may not be included. Information on how the file was last used may not be available either.

name of the last file user
: identity of the last file user who accessed the file using a particular type of access.

CCS-NAME
: Name of the CCS used to encode the file.

RECORD-FORMAT
: contains the format of the records transferred. The field can be assigned the following values:

v	Variable length records
f	Fixed length records
u	No defined record length or the record length is hidden in the transmission format, e.g. records are terminated with a CRLF (Carriage Return Line Feed).

RECORD-SIZE
: contains the maximum length of the records to be transferred.

FILE-AVAILABILITY
: The field can be assigned the following values:

i	File available immediately (immediate).
d	File not available immediately (deferred). The partner is responsible for interpreting the term *deferred*. In the case of openFT partners on BS2000 or z/OS, this means that the file has been migrated.

MAX-FILESIZE
: contains the maximum possible file size in bytes (FTAM-specific value). This value is only as precise as the value returned by the partner system. Since files are created differently in different systems, different values can be displayed for files of the same size.

LEGAL-QUALIFICATION
: contains a legal qualification for the file (corresponds to a copyright, FTAM specific).

5.25 ftshwa - Display admission sets

ftshwa stands for "show admission set", and allows you to examine admission sets.

As a user, you can call *ftshwa* to view your own admission set as well as the standard admission set.

It outputs the following information:

- what limit values the owner of the user ID has set for the individual basic functions
- what limit values the FTAC administrator has set for the user ID for the individual basic functions,
- whether or not the admission set has the FTAC privilege (i.e. if the owner of the admission set is the FTAC administrator).
- whether or not the admission set has the ADM privilege (i.e. if the owner of the admission set is the ADM administrator).

Format

ftshwa -h |
[<user ID 1..32> | @a | @s][-csv]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

user ID | **@a** | **@s**
specifies the user ID for which the admission set is to be displayed.

user ID
You can specify only your own login name here if you are a non-privileged user.

@a for *user ID*
displays information on your admission set and the standard admission set.

@s for *user ID*
returns information only on the standard admission set.

If you specify a non-existent login name, the current standard admission set is displayed for this login name.

user ID not specified
: FTAC displays information on the admission set of the login name under which *ftshwa* was entered.

-csv Specifying *-csv* indicates that the FT admission sets are to be output in the CSV format. The values in the output are separated by semicolons.

-csv not specified
: The FT admission sets are output in the standard format.

Example

Display of command `ftshwa␣@a`:

```
ftshwa @a
                MAX. USER LEVELS              MAX. ADM LEVELS          ATTR
 USER-ID  OBS  OBR  IBS  IBR  IBP  IBF  OBS  OBR  IBS  IBR  IBP  IBF
 *STD     100  100  100  100  100  100  100  100  100  100  100  100
 smith     90   90    0    0    0   90  100* 100* 100* 100* 100* 100*
```

The displayed information has the following meaning:

USER-ID
: The USER-ID column contains the login names to which the respective admission sets belong. If a login name longer than 8 characters is specified, the first 7 characters are output followed by an asterisk (*).

MAX. USER LEVELS / MAX. ADM LEVELS
: The six columns under MAX. USER LEVELS show the values specified by each of these FTAC users for their respective admission sets. The six columns under MAX. ADM LEVELS contain the values set by the FTAC administrator.
The lower of the two values determines whether or not the owner of this admission set may use the basic function specified.

The names of the basic functions are abbreviated as follows:

```
OBS = OUTBOUND-SEND
OBR = OUTBOUND-RECEIVE
IBS = INBOUND-SEND
IBR = INBOUND-RECEIVE
IBP = INBOUND-PROCESSING
IBF = INBOUND-FILE-MANAGEMENT
```

The values in the admission set have the following meaning:

0	The basic function is disabled.
1..99	The basic function is only released for partner systems with the same or a lower security level. You can use the *ftshwptn* command to display a partner system's security level.
100	The inbound basic function is enabled for all partner systems.

An asterisk '*' after the value indicates that this entry was taken from the standard admission set and will automatically be modified if the value in the standard admission set is changed.

ATTR

This column indicates administrator privileges and is empty for non-privileged users.
PRIV in the ATTR column indicates the privileged admission set, i.e. the FTAC administrator.

ADMPR in the ATTR column indicates the ADM administrator.*root*

5.26 ftshwf - Display the attributes of a local file

The command is above all useful in connection with FTAM partners. For openFT partners, information about *binary-fixed* file can be displayed.

With *ftshwf*, you can display the FTAM attributes of a file in the local system. Thus, you can define the file attribute values for file transfer and file management requests involving FTAM partners.

There are three options for outputting the attributes:

– Display the file name
– Display standard information
– Display all attributes of the file

Output is written to standard output.

A precise description of standard output and detailed output can be found in the section “Description of file attribute display” on page 260.

Format

```
ftshwf  -h |
          <file name 1..512>
        [ -s | -l ][ -csv ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

file name
Indicates the file whose attributes are to be displayed. Some of the attributes displayed only apply for FTAM partners who wish to transfer files with openFT-FTAM.

-s Only the file name is output (short).

-l All information available on the file in the partner system is output.

neither *-s* nor *-l* specified:
The standard information is displayed. The amount of information and the layout of the output are described in the section “Description of file attribute display” on page 260.

-csv You use *-csv* to specify that the file attributes are to be output in CSV format. The values are output separated by semicolons. If *-csv* is specified then output is always complete (in the same way as for *-l*) irrespectively of whether *-l* is specified simultaneously or not.

Examples

1. You wish to output the standard scope of information on the *locfile* file on the local system.

```
ftshwf␣locfile

*ripxeacd--- john    214 Apr 30 11:55   /home/john/locfile
```

2. You wish to output detailed information on the FTAM attributes of the *locfile* file on the local system.

```
ftshwf␣locfile␣-l

FILENAME=/home/john/locfile
CRE   otto
MOD   DATE=Apr 28 15:54
REA   DATE=Apr 30 09:01
ATM   DATE=Apr 28 15:54
FILE-AVAILABILITY=i
ACCESS-RIGHTS=ripxeacd---     FILESIZE=214
```

3. Example of a file with the attribute *binary fixed* that is evaluated for openFT partners, see the command *ftmodf* on page 225:

```
ftshwf␣binfix.06␣-l

FILENAME=/home/special/binfix.06
CRE   special
MOD   DATE=Nov 28 15:54
REA   DATE=Dez 05 10:01
ATM   DATE=Dez 05 15:54
BINARY-FILE  RECORD-FORMAT=f  RECORD-SIZE=14156
FILE-AVAILABILITY=i
ACCESS-RIGHTS=ripxeacd---     FILESIZE=42468
```

5.27 ftshwi - Display information on instances

The *ftshwi* command allows you to display information on the openFT instances.

Format

ftshwi -h | [-l | -d] [<instance 1..8> | @a]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-l (long) Detailed information is output, consisting of the instance name, the host name and the instance directory.

-d Displays only the instance directory.

If neither *-l* nor *-d* are set, only the instance name is displayed.

instance | **@a**

Name of the instance on which you want information to be displayed. Instance names have a maximum length of 8 characters and must consist of alphanumeric characters. The first character must not be a number.

@a for *instance*

Information on all instances is output.

If neither an instance name nor *@a* is specified, information is displayed on the instance that is currently set.

Examples

1. You enter *ftshwi* immediately after installation:

```
ftshwi -l @a
Instance Address          Directory
-------- -------          -----------
std      -                /var/openFT/std
```

The output "-" under *Address* means that the default instance logs into all addresses of the system and listens on all addresses.

2. You enter *ftshwi* after the FT administrator has assigned the default instance the address MAPLE using the *ftmodi* command:

```
ftshwi -l @a
Instance Address           Directory
-------  -------           -----------
std      MAPLE             /var/openFT/std
```

   The default instance only logs into the address MAPLE and also only listens on this address.

3. You enter *ftshwi* in a cluster configuration with several instances:

```
ftshwi -l @a
Instance Address             Directory
-------  -------             -----------
maple    CL_MAPLE           /sha_MAPLE/oFT
beech    CL_BEECH           /sha_BEECH/oFT
std      MAPLE              /var/openFT/std
```

Messages of the ftshwi command

If *ftshwi* could not be executed properly, a self-explaining message is output. The exit code is not equal zero in this case.

5.28 ftshwl - Display log records

With *ftshwl*, you can obtain information on all openFT requests logged up to now by openFT.

You can display all log records entered under your own login name.

The log records are marked as FT, FTAC and ADM log records respectively, which means that you can determine the type of log record from the output.

For every request, there is an FTAC log record in which you can find the result of the FTAC admission check. For transfer requests, openFT logs whether it can actually execute this request in FT log records and for remote administration requests in ADM log records. The following applies:

– If FTAC rejects a transfer request as a result of a negative access check, only an FTAC log record exists, and no FT log record.

– An FT log record is only written after the file has been successfully created or opened. If, for instance, it is not possible to find a file, no FT log record is written.

If no options are specified, openFT outputs the current log record. If options are specified, openFT outputs all log records up to the time specified in the command in reverse chronological order, i.e. starting from the most recent record to the oldest record.

There are three types of output: short output, long output and CSV output (**C**omma **S**eparated **V**alue).

Output is written to standard output.

Format

ftshwl -h |
[<user ID 1..32> | @a]
[-rg=[[[[yyyy]mm]dd]hhmm|#1..999999999999|0..999|:0..999][-
[[[[yyyy]mm]dd]hhmm|#1..999999999999|0..999|:0..999]]]
[-rt=[t][c][a]]
[-ff=[t][m][r][d][a][C][D][M][I][f]]
[-ini=I | -ini=r | -ini=Ir | -ini=rI]
[-pn=<partner 1..200>]
[-fn=<file name 1..512>]
[-nb=1..99999999 | -nb=@a]
[-rc=0..ffff | -rc=@f]
[-l][-csv]
[-tid=1..2147483647]
[-adm=<administrator id 1..32>]
[-ri=<routing info 1..200>]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

user ID | **@a**
is used to specify the login name(s) for which log records are to be displayed. As ordinary user, you can only specify your own login name.

@a for *user ID*
This also displays information, but only on the log records that refer to your own login name.

user ID not specified
Only the log records for the login name under which the command was entered are displayed.

-rg=[[[[yyyy]mm]dd]hhmm]**-**[[[[yyyy]mm]dd]hhmm]
You can *-rg* to specify the start and/or end of a logging interval.

[[[yyyy]mm]dd]hhmm
When specifying a time, a 4-digit specification is interpreted as the time expressed in hours and minutes, a 6-digit specification as the day (date) and time in hours and minutes, an 8-digit specification as the month, day, and time in hours and minutes, and a 12-digit

specification as the year, month, day, and time in hours and minutes. The largest possible value that can be specified as the date is 20380119 (January 19, 2038).
openFT then displays all the log records written during the specified time period. The older time is taken to be the start time and the earlier time as the end time.
The optional data (*[[[yyyy]mm]dd]*) is automatically replaced by current values.

If you omit the limit after the dash, the current time is taken. If you omit the limit before the dash, the time of the first log record written is taken.

-rg=[[[yyyy]mm]dd]hhmm
If the minus sign is missing, the range is the exact minute specified. The largest possible value that can be specified as the date is 20380119 (January 19, 2038). The optional data (*[[[yyyy]mm]dd]*) is automatically replaced by current values.

-rg=[**#**1..999999999999]**-**[**#**1..999999999999]
-rg is used to specify the start and/or end of a range of log IDs.

#1..999999999999
The selection of a log ID is indicated by the leading # character. openFT then displays all the log records which lie within the specified range.

If the log ID limit before the dash is omitted, the current ID is taken, and if the log ID limit after the dash is omitted, the ID of the first log record written is taken.

-rg=#1..999999999999
If the minus sign is omitted, the range is restricted to the specified log ID only.

-rg=[0..999][**-**[0..999]]
Here you specify with *-rg* a relative time period as a multiple of 24 hours (i.e. as a number of days). Note that the relative time period is calculated with an accuracy of one second from the current time. You have the following options (*d1* and *d2* 1 through 3 digits):

– *-rg=d1-d2* outputs all log records that are between *d1* and *d2* days old, irrespective of whether *d1* is larger or smaller than *d2*.

– *-rg=d1-* outputs all log records that are no more than *d1* days old.

– *-rg=-d2* outputs all log records that are at least *d2* days old.

-rg=[**:**0..999][**-**[**:**0..999]]

Here you specify with *-rg* a relative time period in minutes. You have the following options in this case (*m1* and *m2* 1 through 3 digits):

- *-rg=m1-:m2* outputs all log records that are between *m1* and *m2* minutes old, irrespective of whether *m1* is larger or smaller than *m2*.
- *-rg=:m1* (or *-rg=:m1-*) outputs all log records that are no more than *m1* minutes old.
- *-rg=-:m2* outputs all log records that are at least *m2* minutes old.

-rg not specified

The range is not a selection criterion.

-rt=[**t**][**c**][**a**]

Defines which type of log record is to be displayed.
You may specify *t*, *c*, *a* and any combination of these values:

t The FT log records are displayed.

c The FTAC log records are displayed.

a The ADM log records are displayed. For further details, refer to the openFT manual "Installation and Administration".

-rt not specified

The record type is not a selection criterion.

-ff=[**t**][**m**][**r**][**d**][**a**][**C**][**D**][**M**][**l**][**f**]

Defines the FT function for which log records are to be output. Possible values are: *t*, *m*, *r*, *d*, *a*, *C*, *D*, *M*. *l*, *f* or any combination of these values. The entries *m*, *r*, *d*, *a*, *C*, *D*, *M* and *l* are only reasonable for FTAC log records. The entry *f* is only reasonable for ADM log records. *t* is reasonable for all log records.

t All log records for the function "transfer files" are output.

m All log records for the function "modify file attributes" are output.

r All log records for the function "read directories" are output.

d All log records for the function "delete files" are output.

a All log records for the function "read file attributes" are output.

C All log records for the function "Create directory" are output.

D All log records for the function "Delete directory" are output.

M All log records for the function "Modify directory" are output.

I All log records for the function "inbound FTP access" are output. These log records are written if incorrect admission data (FTAC transfer admission or user ID/password) was specified for inbound FTP access.

f This specification is only of significance to the administrator of the remote administration server.

-ff not specified
The FT function is not a selection criterion.

-ini=l | **-ini=r** | **-ini=lr** | **-ini=rl**
Defines the initiator for which log records are to be output. Possible values are: *l*, *r*, *lr*, *rl*.

l (local) Only log records belonging to openFT requests issued locally are output.

r (remote) Only log records belonging to openFT requests issued remotely are output.

lr, rl The log records belonging to openFT requests issued locally and remotely are output.

-ini not specified
The initiator is not a selection criterion.

-pn=partner
Defines the partner system to which the log records are to be output. Partner is the name of the partner in the partner list or the address of the partner system. For details on address specifications, see section “Defining the partner computer” on page 90.

-pn not specified
The partner system is not a selection criterion.

-fn=file name
Defines the file to which the log records are to be output. You can specify wildcards such as "*" (asterisk, i.e. any character string) and "?" (question mark, i.e. single character).

-fn not specified
The file name is not a selection criterion.

-nb=number | **@a**
Defines the number of log records to be output.

@a for *number*
All log records are output.

-nb not specified
If *-rg* has also been specified, *-nb* is replaced by the value *-nb=@a*.

If *-rg* is also not specified, *-nb* is replaced by the value *-nb=1*.

-rc=0..ffff | **@f**
Defines the reason code as a selection criterion for log record output.

0 .. ffff
All log records with a specified reason code are output.

@f All log records with reason codes other than 0000 are output. This criterion yields a list of log records for all requests terminated with error messages.

-rc not specified
The reason code is not a selection criterion.

-l Defines that the log records are to be output in long form.

-l not specified
The log records are output in short form if *-csv* has not been specified.

-csv You can use *-csv* to specify that the log records are to be output in the CSV format. The values in the output are separated by semicolons. If *-csv* is specified, output is always in long form (analogous to *-l*) regardless of whether or not *-l* has also been specified.

-csv not specified
The log records are output in the standard format, i.e. in abbreviated form if *-l* is not specified and in detailed form if *-l* is specified.

-tid=request id
-tid specifies the request number for which you want to output the log records.

-tid not specified
The request id is not a selection criterion.

-adm=administrator id
-adm specifies the administrator ID for which you want to output the ADM log records.

-adm not specified
: The administrator id is not a selection criterion.

-ri=routing info
: *-ri* specifies the routing information for which you want to output the ADM log records.

-ri not specified
: The routing info is not a selection criterion.

Examples

1. All log records that are more than two days (48 hours) old are output:

   ```
   ftshwl -rg=-2
   ```

2. All log records that are more than 15 minutes old but less than 30 minutes old are output:

   ```
   ftshwl  rg=:15-:30
   ```

3. All log records that are less than 30 minutes old are output:

   ```
   ftshwl -rg=:30
   ```

4. All log records that are more than 30 minutes old are output:

   ```
   ftshwl -rg=-:30
   ```

5. The last 10 log records where FTAC checks failed (reason code not equal to 0) are output:

   ```
   ftshwl -rc=@f -rt=c -nb=10
   ```

5.28.1 Description of log record output

Log records can be displayed using the openFT Explorer or by using the *ftshwl* command. You can choose between a short overview, detailed information or, if further processing is to be performed with external programs, output in the CSV format.

The log records are identified by log IDs. The log IDs are assigned in ascending order, but for technical reasons the numbering is not contiguous (i.e. there may be gaps).

5.28.1.1 Logging requests with preprocessing/postprocessing

For security reasons, only the first 32 characters (or 42 characters in the case of *ftexecsv* preprocessing) of a preprocessing or postprocessing command are transferred to the log record. By arranging the call parameters appropriately or by inserting blanks, you can influence which command parameters do not appear in the log.

5.28.1.2 Short output format of a FT or FTAC log records

Example

The option *-rt=tc* causes only FT and FTAC log records to be output.

```
$ftshwl -rt=tc -nb=12
TYP LOG-ID TIME     RC    PARTNER  INITIAT. PROFILE  USER-ADM FILENAME
2009-05-05
CA    8273 09:16:07 0000 >PARTLINU *REMOTE  pr1      user1    file.10
CA    8272 09:16:07 0000 >PARTLINU user1             user1
CD    8271 09:15:30 0000 <PARTLINU *REMOTE  pr1      user1    file.new
CD    8270 09:15:30 0000 <PARTLINU user1             user1
CM    8269 09:15:03 0000 <PARTLINU *REMOTE  pr1      user1    file.rem
CM    8268 09:15:03 0000 <PARTLINU user1             user1    file.new
CR    8267 09:14:14 0000 >PARTLINU *REMOTE  pr1      user1    .
CR    8266 09:14:14 0000 >PARTLINU user1             user1
T     8265 09:13:50 0000 >PARTLINU user1             user1    file.10
T     8264 09:13:50 0000 <PARTLINU *REMOTE           user1    file.rem
C     8263 09:13:49 0000 <PARTLINU *REMOTE  pr1      user1    file.rem
C     8262 09:13:49 0000 >PARTLINU user1             user1    file.10
```

Explanation

TYP Comprises three columns. The first column specifies whether the log record is an FT or FTAC log record:

- T FT log record
- C FTAC log record

The second and third column identify the FT function:

- ␣ (empty): transfer file
- A read file attributes (only in the FTAC log record)
- D delete file (only in the FTAC log record)
- C create file (only in the FTAC log record)
  possible only for transfer requests issued in the remote partner system
- M modify file attributes (only in the FTAC log record)
- R read directory (only in the FTAC log record)
- CD create directory (only in FTAC log record)
- DD delete directory (only in FTAC log record)
- MD modify directory attributes (only in FTAC log record)
- L Login: Failed inbound FTP access (only in FTAC log record)

LOG-ID
Log record number

TIME
specifies time when the log record was written

RC Reason code. Specifies whether a request was successful (RC=0) or if not, why it was rejected or cancelled. Additional information on the reason code is available using the *fthelp* command.

PARTNER
Provides information about the partner system involved. The name in the partner list or the address of the partner system, possibly truncated to 8 characters, or the name under which the partner system is entered in the TNS is output.

The name or address of the partner system is preceded by an identifier to indicate the direction of the request.

> The request is sent to partner system. This transfer direction is specified for a
>
> – send request
> – a request to display file attributes
> – a request to display directories

< The request is sent to local system. This transfer connection is specified for

– a receive request
– a request to modify file attributes
(When a FTAM partner modifies the access rights of a local file, two log records are written. No direction is specified in front of PARTNER in this case.)
– a request to delete files

INITIAT.
Request initiator. If initiated in the remote system: *REMOTE.

PROFILE
Name of the profile used for file transfer (only in FTAC log record).

USER-ADM
Login name to which the requests in the local system refer.

If a login name longer than 8 bytes was specified, the first seven bytes are output, followed by an asterisk (*).

FILENAME
Local file name

5.28.1.3 Long output format of an FT log record

Example

```
ftshwl @a -rg=#28 -l
LOGGING-ID  = 28        RC      = 2156        TIME     = 2009-05-23 10:53:22
   TRANS    = TO        REC-TYPE= FT          FUNCTION = TRANSFER-FILE
   PROFILE  =           PCMD    = NONE        STARTTIME= 2009-05-23 10:53:20
   TRANS-ID = 65539     WRITE   = REPLACE     REQUESTED= 2009-05-23 10:53:20
   TRANSFER =         0 kB                    CCS-NAME = ISO88591
   SEC-OPTS = ENCR+DICHK, RSA-768 / AES-128
   INITIATOR= smtih
   USER-ADM = smtih
   PARTNER  = FTSERV01
   FILENAME = test01
   ERRINFO  = CreateDirectory: Access denied
```

Explanation

LOGGING-ID
: Log record number; up to twelve characters in length

TRANS
: Transfer direction

  TO Transfer direction to the partner system. This transfer direction is specified for

  – a send request
  – a request to display the file attributes
  – a request to display the directories

  FROM
  : Transfer direction to the local system. This transfer direction is specified for

    – a receive request
    – a request to modify the file attributes
    – a request to delete files

PROFILE
: Name of profile used

TRANS-ID
: Request number

TRANSFER
: Number of bytes transferred

SEC-OPTS
: Security options used during transfer

  ENCR Encryption of the request description

  DICHK Data integrity check of the request description

  DENCR Encryption of the transferred file content

  DDICHK Data integrity check of the transferred file content

  LAUTH Authentication of the local system in the remote system

  RAUTH Authentication of the remote system in the local system

  RSA-nnn
  : Length of the RSA key used for the encryption

  AES-128 / AES-256 / DES
  : The encryption algorithm used

INITIATOR
: Request initiator. If initiated in the local system: login name. If initiated in the remote system: *REMOTE

USER-ADM
: Login name to which the requests in the local system refer

PARTNER
: Identifies the partner system in question.
  The name in the partner list or the address of the partner system, possibly truncated to 8 characters, or the name under which the partner system is entered in the TNS is output.
  In the case of requests issued from a remote computer, it is also possible for *%strange* to be output followed by a part of the address of the partner system if the partner system is not entered in the TNS and TCP/IP-RFC1006 was not used as the transport system.

FILENAME
: Local file name

ERRINFO
: Additional information on the error message if an error occurred during a transfer.

RC Reason code. Specifies whether a request was successful (RC=0) or if not, why it was rejected or cancelled. You can obtain further information with the *fthelp* command.

REC-TYPE
: Specifies whether the log record is an FT log record.

PCMD
: Indicates whether follow-up processing was specified and started. Possible values:

  NONE
  : No follow-up processing specified

  STARTED
  : Follow-up processing was started (contains no information about the successful completion of follow-up processing!).

  NOT-STARTED
  : Follow-up processing could not be started.

WRITE
: Write mode. The field is assigned a value only for outbound requests; for inbound requests, it contains a blank. Possible values:

  NEW A new file is created. If a file with this name already exists, file transfer is aborted.

  EXT An existing file is extended, otherwise a new is created.

  REPLACE
  : An existing file is overwritten. If it does not already exist, it is created.

TIME
: Specifies time when log record was written

FUNCTION
: FT function

  TRANSFER-FILE
  : Transfer file

STARTTIME
: Indicates the start time of the request, if this is was specified explicitly by the initiator on the remote system; otherwise, this field is empty.

STORETIME
: If the request was submitted in the remote system then the time of the entry in the request queue is displayed here.

REQUESTED
: When initiative in the local system, the time of issue of the request is shown here.

CCS-NAME
: Name of the character set used to code the local file.

5.28.1.4 Long output format of an FTAC log record

Example

```
ftshwl @a -rg=#5172 -l
LOGGING-ID  = 00005172  RC      = 0000        TIME     = 2009-04-03 09:38:06
   TRANS    = TO        REC-TYPE= FTAC        FUNCTION = TRANSFER-FILE
   PROFILE  = remadmin  PRIV    = NO
   INITIATOR= *REMOTE
   USER-ADM = thomasw
   PARTNER  = angel.domain1.de
   FILENAME = |ftexecsv ftshwo -tn -a -u -ccs=IS088591
```

Explanation

LOGGING-ID
: Log record number, up to twelve characters in length

TRANS
: Transfer direction

  TO
  : Transfer direction to partner system. This transfer direction is specified for

    – a send request
    – a request to display the file attributes
    – a request to display the directories

  FROM
  : Transfer direction to local system. This transfer direction is specified for

    – a receive request
    – a request to modify the file attributes
    – a request to delete files

  BOTH
  : The request direction is to the partner system and to the local system. When an FTAM partner modifies the access rights of a local file, two log records are written. The direction BOTH is specified in each.

PROFILE
: Name of the profile used

INITIATOR
: Request initiator. If initiated in the local system: login name. If initiated in the remote system: *REMOTE

USER-ADM
: Login name to which the requests in the local system refer

PARTNER
: Identifies the partner system in question.
The name in the partner list or the address of the partner system, possibly truncated to 8 characters, or the name under which the partner system is entered in the TNS is output.
In the case of requests issued from a remote computer, it is also possible for *%strange* to be output followed by a part of the address of the partner system if the partner system is not entered in the TNS and TCP/IP-RFC1006 was not used as the transport system.

FILENAME
: Local file name

RC
: Reason code. Specifies whether a request was successful (RC=0) or if not, why it was rejected or cancelled. You can use the *fthelp* command to obtain further information.

REC-TYPE
: Specifies whether the log record is an FTAC log record.

PRIV
: Specifies whether or not the FT profile being used is privileged

TIME
: Specifies time when the log record was written

FUNCTION
: FT function

  TRANSFER-FILE
  : Transfer file

  READ-FILE-ATTR
  : Read file attributes

  DELETE-FILE
  : Delete file

  CREATE-FILE
  : Create file (possible only in requests submitted in the remote partner system)

  MODIFY-FILE-ATTR
  : Modify file attributes

READ-FILE-DIR
: Read directories

CREATE-FILE-DIR
: Create file directory

DELETE-FILE-DIR
: Delete file directory

MODIFY-FILE-DIR
: Modify file directory

LOGIN
: Login: Inbound FTP access.
This log record is written if incorrect admission data was specified for inbound FTP access.

5.28.2 Reason codes of the logging function

The FTAC log records contain a reason code which indicates whether an request was accepted after the admission check successfully and if not, why it was rejected.

You can use the *fthelp* command to output the message text associated with the code number (see page 207):

fthelp *code-number*

In many codes, the last three digits correspond to the number of the associated openFT message.

In addition, there are a certain number of codes which do not correspond to openFT messages. These are listed in the table below:

RC	Reason
0000	Request successfully completed.
1001	Request rejected. Invalid transfer admission
1003	Request rejected. Transfer direction not permissible
1004	Request rejected. Illegal partner
1006	Request rejected. Violation of file name restriction
100f	Request rejected. Violation of success processing restriction
1010	Request rejected. Violation of failure processing restriction

RC	Reason
1011	Request rejected. Violation of write mode restriction
1012	Request rejected. Violation of FT function restriction
1014	Request rejected. Violation of data encryption restriction
2001	Request rejected. Syntax error on file name extension
2004	Request rejected. Overall length of follow-up processing exceeds 1000 characters
3001	Request rejected. Invalid user identification
3003	Request rejected. Invalid password
3004	Request rejected. Transfer admission locked
3011	Request rejected. Violation of user outbound send level
3012	Request rejected. Violation of user outbound receive level
3013	Request rejected. Violation of user inbound send level
3014	Request rejected. Violation of user inbound receive level
3015	Request rejected. Violation of user inbound processing level
3016	Request rejected. Violation of user inbound file management level
3021	Request rejected. Violation of ADM outbound send level
3022	Request rejected. Violation of ADM outbound receive level
3023	Request rejected. Violation of ADM inbound send level
3024	Request rejected. Violation of ADM inbound receive level
3025	Request rejected. Violation of ADM inbound processing level
3026	Request rejected. Violation of ADM inbound file management level

5.29 ftshwm - Display monitoring values of openFT operation

The *ftshwm* command allows you to output the current monitoring values from openFT operation. In order to do this, the FT administrator must have activated monitoring (*ftmodo -mon=n* command) and the asynchronous openFT server must be running.

Format

ftshwm -h |
[-ty]
[-raw]
[-po=<polling interval 1..600> [-pnr=<polling number 1..3600>]]
[-csv]
[<name 1..12> [... <name(100) 1..12>]| @a]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-ty The types and scaling factors are to be output in place of the monitoring values and metadata.

The metadata type can be *TIME (timestamp) or *STRING (text output of the chosen selection).

A monitoring value can have one of the following types:
INT, BOOL or PERCENT (integer, on/off value or percentage). In the case of integer values, the scaling factor may be specified in brackets: INT(<scaling factor>).

The scaling factor of a monitoring value is only significant for output in CSV format. In this case, it is the number by which the value shown must be divided in order to obtain the real value.

-raw must not be specified at the same time.

-raw Monitoring values are to be output as unedited raw data. This option is intended to be used in conjunction with external programs for further processing. The option must not be specified in conjunction with *-ty*. Monitoring values of the object *Duration* are not output.

If the specification is not used, the data is output in print-edited form.

The following section “Description of the monitoring values” on page 292 contains a table with notes that show what values are output when the *-raw* option is specified or is not specified and how the values are to be interpreted depending on this option.

-po=polling interval
: Data is to be output initially after the specified polling interval in seconds has elapsed and then repeated at this interval.

  If you also specify *-pnr*, you can limit the number of times the data is output. If you specify *-po* without *-pnr*, output is repeated an unlimited number of times.

  If repeated output has been started with the *-po* option (with or without *-pnr*), it can be cancelled by an interrupt signal. Output is also cancelled in the event of an error, when openFT is terminated, or when monitoring is terminated.

  Possible values: 1 through 600.

-po not specified
: The monitoring values are output immediately and once only.

-pnr=polling number
: *-pnr* specifies the number of times data is output. *-pnr* can only be specified in conjunction with *-po*.

  Possible values: 1 through 3600.

-csv The information is to be output in CSV format. First, the short names of the monitoring values are output in one row as the field names. This is followed by a row containing the monitoring values or their types and scaling factors as decimal numbers.

  You can limit the scope of the output by specifying individual monitoring values that are significant for you.

name [name ...] | **@a**
: The specified monitoring value or, if *-ty* is specified, the type and scaling factor associated with the named value is to be output.

  name must be one of the short names of the monitoring values as they appear in the CSV header. You can specify up to 100 names separated by blanks.

  @a for *name*
  : All openFT monitoring values or the types and scaling factors of all openFT monitoring values are to be output.

name not specified

A predefined default set of monitoring values is output (see the section “Description of the monitoring values” on page 292).

5.29.1 Description of the monitoring values

The table below shows all the monitoring values output with the option *@a*. You can instead specify a list of any of the monitoring values shown in the table.

The first two letters of the name indicate the data object that the monitoring value belongs to:

– Th = Throughput
– Du = Duration
– St = State

The second component of the name indicates the performance indicator, e.g. *Netb* for net bytes. In the case of monitoring values for the *Throughput* or *Duration* data object, the last 3 letters of the name indicate the types of requests from which the monitoring value originates, e.g.

– Ttl = FT Total
– Snd = FT Send requests
– Rcv = FT Receive requests
– Txt = Transfer of text files
– Bin = Transfer of binary files
– Out = FT Outbound
– Inb = FT Inbound

If monitoring is deactivated for all partners (*ftmodo -monp=*), only the following values are populated:

Status: StCLim, StCAct, StRqLim, StRqAct, StOftr, StFtmr, StFtpr, StTrcr

All the other values are set to 0.

Name	Meaning	Output with @a only	Output unit	
			Formatted	Raw
ThNetbTtl	Throughput in net bytes: Number of bytes transferred		Number of bytes per second	Bytes, accumu-lated

Name	Meaning	Output with @a only	Output unit	
			Formatted	Raw
ThNetbSnd	Throughput in net bytes (send requests): Number of bytes transferred with send requests		Number of bytes per second	Bytes, accumulated
ThNetbRcv	Throughput in net bytes (receive requests): Number of bytes transferred with receive requests		Number of bytes per second	Bytes, accumulated
ThNetbTxt	Throughput in net bytes (text files): Number of bytes transferred when transferring text files	x	Number of bytes per second	Bytes, accumulated
ThNetbBin	Throughput in net bytes (binary files): Number of bytes transferred when transferring binary files	x	Number of bytes per second	Bytes, accumulated
ThDiskTtl	Throughput in disk bytes: Number of bytes read from files or written to files with transfer requests		Number of bytes per second	Bytes, accumulated
ThDiskSnd	Throughput in disk bytes (send requests): Number of bytes read from files with send requests		Number of bytes per second	Bytes, accumulated
ThDiskRcv	Throughput in disk bytes (receive requests): Number of bytes written to files with receive requests		Number of bytes per second	Bytes, accumulated
ThDiskTxt	Throughput in disk bytes (text files): Number of bytes read from text files or written to text files with transfer requests	x	Number of bytes per second	Bytes, accumulated
ThDiskBin	Throughput in disk bytes (binary files): Number of bytes read from binary files or written to binary files with transfer requests	x	Number of bytes per second	Bytes, accumulated

Name	Meaning	Output with @a only	Output unit	
			Formatted	Raw
ThRqto	openFT requests: Number of openFT requests received		Number per second	Number, accumulated
ThRqft	File transfer requests: Number of file transfer requests received	x	Number per second	Number, accumulated
ThRqfm	File management requests: Number of file management requests received	x	Number per second	Number, accumulated
ThSuct	Successful requests: Number of successfully completed openFT requests		Number per second	Number, accumulated
ThAbrt	Aborted requests: Number of aborted openFT requests		Number per second	Number, accumulated
ThIntr	Interrupted requests: Number of interrupted openFT requests		Number per second	Number, accumulated
ThUsrf	Requests from non-authorized users: Number of openFT requests in which the user check was terminated with errors		Number per second	Number, accumulated
ThFoll	Follow-up processing operations started: Number of follow-up processing operations started	x	Number per second	Number, accumulated
ThCosu	Connections established: Number of connections successfully established	x	Number per second	Number, accumulated
ThCofl	Failed connection attempts: Number of attempts to establish a connection that failed with errors		Number per second	Number, accumulated
ThCobr	Disconnections: Number of disconnections as a result of connection errors		Number per second	Number, accumulated

Name	Meaning	Output with @a only	Output unit	
			Formatted	Raw
DuRqtlOut	Maximum request duration Outbound: Maximum request duration of an outbound request	x	Milliseconds [1]	-
DuRqtlInb	Maximum request duration Inbound: Maximum request duration of an inbound request	x	Milliseconds [1]	-
DuRqftOut	Maximum request duration Outbound transfer: Maximum duration of an outbound file transfer request	x	Milliseconds [1]	-
DuRqftInb	Maximum request duration Inbound transfer: Maximum duration of an inbound file transfer request	x	Milliseconds [1]	-
DuRqfmOut	Maximum request duration Outbound file management: Maximum duration of an outbound file management request	x	Milliseconds [1]	-
DuRqfmInb	Maximum request duration Inbound file management: Maximum duration of an inbound file management request	x	Milliseconds [1]	-
DuRqesOut	Maximum outbound request waiting time: Maximum waiting time before an outbound request is processed (for requests without a specific start time)	x	Milliseconds [1]	-
DuDnscOut	Maximum duration of an outbound DNS request: Maximum time an outbound openFT request was waiting for partner checking	x	Milliseconds [1]	-

Name	Meaning	Output with @a only	Output unit	
			Formatted	**Raw**
DuDnsclnb	Maximum duration of an inbound DNS request: Maximum time an inbound openFT request was waiting for partner checking	x	Milliseconds [1]	-
DuConnOut	Maximum duration of establishment of a connection: Maximum time between requesting a connection and receiving confirmation of a connection for an outbound openFT request	x	Milliseconds [1]	-
DuOpenOut	Maximum file open time (outbound): Maximum time an outbound openFT request required to open the local file	x	Milliseconds [1]	-
DuOpenInb	Maximum file open time (inbound): Maximum time an inbound openFT request required to open the local file	x	Milliseconds [1]	-
DuClosOut	Maximum file close time (outbound): Maximum time an outbound openFT request required to close the local file	x	Milliseconds [1]	-
DuClosInb	Maximum file close time (inbound): Maximum time an inbound openFT request required to close the local file	x	Milliseconds [1]	-
DuUsrcOut	Maximum user check time (outbound): Maximum time an outbound openFT request required to check the user ID and transfer admission	x	Milliseconds [1]	-

Name	Meaning	Output with @a only	Output unit	
			Formatted	Raw
DuUsrcInb	Maximum user check time (inbound): Maximum time an inbound openFT request required to check the user ID and transfer admission	x	Milliseconds [1]	-
StRqas	Number of synchronous requests in the ACTIVE state		Average value [2]	Current number
StRqaa	Number of asynchronous requests in the ACTIVE state		Average value [2]	Current number
StRqwt	Number of requests in the WAIT state		Average value [2]	Current number
StRqhd	Number of requests in the HOLD state		Average value [2]	Current number
StRqsp	Number of requests in the SUSPEND state		Average value [2]	Current number
StRqlk	Number of requests in the LOCKED state		Average value [2]	Current number
StRqfi	Number of requests in the FINISHED state	x	Average value [2]	Current number
StCLim	Maximum number of connections: Upper limit for the number of connections established for asynchronous requests.		Value currently set	
StCAct	Number of occupied connections for asynchronous requests		Share of StCLim in % [3]	Current number
StRqLim	Maximum number of requests: Maximum number of asynchronous requests in request management		Value currently set	
StRqAct	Entries occupied in request management		Share of StRqLim in % [3]	Current number
StOftr	openFT Protocol activated/deactivated		ON (activated) OFF (deactivated)	

Name	Meaning	Output with @a only	Output unit	
			Formatted	**Raw**
StFtmr	FTAM protocol activated/deactivated		ON (activated) OFF (deactivated)	
StFtpr	FTP protocol activated/deactivated		ON (activated) OFF (deactivated)	
StTrcr	Trace activated/deactivated	x	ON (activated) OFF (deactivated)	

[1] Maximum value of the monitoring interval (= time elapsed since the last time the monitoring values were queried or since the start of monitoring).

[2] Average value of the monitoring interval (= time elapsed since the last time the monitoring values were queried or since the start of monitoring). Format: n.mm, where n is an integer and mm are to be interpreted as decimal places.

[3] If the reference value is reduced in live operation, it is possible for the value output to lie above 100 (%) temporarily.

Example

```
ftshwm

openFT(std)   Monitoring (formatted)
MonOn=2009-02-16 15:36:12 PartnerSel=OPENFT  RequestSel=ONLY-
ASYNC,ONLY-LOCAL
2009-02-17 15:40:01

 Name      Value
 ----------------
 ThNetbTtl 38728
 ThNetbSnd 38728
 ThNetbRcv 0
 ThDiskTtl 16384
 ThDiskSnd 16384
 ThDiskRcv 0
 ThRqto    1
 ThSuct    0
 ThAbrt    0
 ThIntr    0
 ThUsrf    0
 ThCofl    0
 ThCobr    0
 StRqas    0.00
 StRqaa    8.66
 StRqwt    1.66
 StRqhd    0.00
 StRqsp    0.00
 StRqlk    0.00
 StCLim    16
 StCAct    37
 StRqLim   1000
 StRqAct   1
 StOftr    ON
 StFtmr    OFF
 StFtpr    OFF
```

Explanation

The default output format begins with a header containing the following specifications:

– Name of the openFT instance and selected data format (*raw* or *formatted*)
– Monitoring start time and partner and request selection
– Current timestamp

This is followed by the list of default values, see also page 292.

5.30 ftshwo - Display operating parameters

The *ftshwo* command outputs the operating parameters of the local openFT system. Alongside the standard output and output in CSV format, output may also be specified as a platform-specific command sequence. In this way, it is possible to save the settings and then load them onto another computer.

The FT administrator can set or modify the operating parameters with the *ftmodo* command.

The transfer admission of the ADM trap server is not output with the default output format and CSV output format. It only appears as a command sequence in the output (*-px*, *-pw*, *-p2*, *-pz*).

Format

ftshwo -h |
[-csv | -px | -pw | -p2 | -pz]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-csv The operating parameters are output in CSV format. The individual values are separated by semicolons.

-px The operating parameters are output as a command string. This can be called as a shell procedure on Unix systems in order to regenerate the identical operating parameters.

-pw The operating parameters are output as a command string. This can be called as a batch procedure on Windows systems in order to regenerate the identical operating parameters.

-p2 The operating parameters are output as a command string. This can be called as an SDF procedure on BS2000/OSD systems in order to regenerate the identical operating parameters.

-pz The operating parameters are output as a command string. This can be called as a Clist procedure on z/OS systems in order to regenerate the identical operating parameters.

No option specified
The operating parameters are output in standard format.

Example

```
ftshwo
STARTED PROC-LIM CONN-LIM ADM-CLIM RQ-LIM MAX-RQ-LIFE TU-SIZE KEY-LEN CCS-NAME
  YES      NONE       16        8     2000       30      65535    768   ISO88591
PTN-CHK DYN-PART SEC-LEV  FTAC-LOG FT-LOG ADM-LOG     USE TNS
  STD        ON   B-P-ATTR    ALL    ALL    ALL          NO
OPENFT-APPL      FTAM-APPL          FTP-PORT           ADM-PORT         ADM-CS
*STD             *STD               21                 11000               NO
ACTIVE           ACTIVE             ACTIVE             ACTIVE
HOST-NAME        IDENTIFICATION / LOCAL SYSTEM NAME
*NONE            servus / $FJAM,SERVUS

ADM-TRAP-SERVER
*NONE

TRAP: SS-STATE FT-STATE PART-STATE PART-UNREA RQ-STATE TRANS-SUCC TRANS-FAIL
CONS     OFF      OFF        OFF         OFF       OFF        OFF         OFF
ADM               OFF        OFF         OFF       OFF        OFF         OFF

FUNCT:  SWITCH PARTNER-SELECTION    REQUEST-SELECTION    OPTIONS
MONITOR    ON   ALL                ONLY-SYNC,ONLY-LOCAL
TRACE      ON   OPENFT,FTP,ADM      ALL                  NO-BULK-DATA
```

Meaning of the output together with the associated command options:

Field name	Meaning and values	Command/ option
STARTED	Specifies whether the asynchronous openFT server has started (YES) or not (NO).	*ftstart* *ftstop*
PROC-LIM	Maximum number of openFT servers available for the processing of asynchronous requests.	*ftmodo -pl=*
CONN-LIM	Maximum number of asynchronous requests that can be processed simultaneously.	*ftmodo -cl=*
ADM-CLIM	Maximum number of asynchronous administration requests including ADM traps that can be processed simultaneously.	*ftmodo -admcl=*
RQ-LIM	Maximum number of file transfer requests that can simultaneously be present in the local system's request queue.	*ftmodo -rql=*
MAX-RQ-LIFE	Maximum lifetime of requests in the request queue (in days).	*ftmodo -rqt=*
TU-SIZE	Upper limit for message length at transport level (in bytes).	*ftmodo -tu=*
KEY-LEN	Length of the RSA key currently used to encrypt the AES/DES key.	*ftmodo -kl=*
CCS-NAME	Name of the character set used by default for file transfer requests, see page 85	*ftmodo -ccs=*

Field name	Meaning and values	Command/ option
PTN-CHK	Setting for sender verification: ADDR: address STD: identification	*ftmodo -ptc=*
DYN-PART	Setting for dynamic partner entries: ON (activated) OFF (deactivated)	*ftmodo -dp=*
SEC-LEV	Default security level for partners in the partner list for which no security level has been set: 1..100: Fixed security level. 1 is the lowest and 100 the highest security level. B-P-ATTR: The security level is depending on the partner's attributes, i.e.: 10 if the partner has been authenticated. 90 if the partner is known in the transport system. 100 otherwise, i.e. if the partner has only been identified by its address.	*ftmodo -sl=*
FTAC-LOG	Scope of FTAC logging: ALL: All FTAC access checks MODIFY: Modifying file management requests and rejected FTAC access checks REJECTED: Only rejected FTAC access checks	*ftmodo -lc=*
FT-LOG	Scope of FT logging: ALL: All requests FAIL: Only errored FT requests NONE: FT Logging deactivated	*ftmodo -lt=*
ADM-LOG	Scope of ADM logging: ALL: All requests FAIL: Only errored ADM requests MODIFY: only modifying ADM requests NONE: ADM Logging deactivated	*ftmodo -la=*

Field name	Meaning and values	Command/ option
USE TNS	Specifies whether the TNS is to be used (YES) or not (NO).	*ftmodo -tns=*
OPENFT-APPL	Port number of the local openFT server, possibly extended by the transport selector. *STD means that the default value is used i.e. 1100 and $FJAM in Transdata format (EBCDIC, 8 characters long, padded with blanks). Line 2: ACTIVE: openFT protocol activated DISABLED: openFT protocol (inbound) deactivated INACT: openFT protocol (inbound) not available	*ftmodo -openft=* *ftmodo -acta=*
FTAM-APPL	Port number of the local FTAM server, possibly extended by the transport selector, the session selector and the presentation selector. *STD means that the default value is used i.e. 4800 and $FTAM in Transdata format (EBCDIC, 8 characters long, padded with blanks) Line 2: ACTIVE: FTAM protocol activated DISABLED: FTAM protocol (inbound) deactivated INACT: FTAM protocol (inbound) not available NAVAIL: FTAM not installed	*ftmodo -ftam=* *ftmodo -acta=*
FTP-PORT	Port number used by local FTP server. Default port: 21 Line 2: ACTIVE: FTP protocol activated DISABLED: FTP protocol (inbound) deactivated INACT: FTP protocol (inbound) not available NAVAIL: FTP not installed	*ftmodo -ftp=* *ftmodo -acta=*
ADM-PORT	Port number used by remote administration. Default port: 11000	*ftmodo -adm=*

Field name	Meaning and values	Command/ option
	Line 2: ACTIVE: remote administration activated DISABLED: remote administration (inbound) deactivated INACT: remote administration (inbound) not available	*ftmodo -acta=*
ADM-CS	Specifies whether the local openFT instance is flagged as a remote administration server (YES) or not (NO).	*ftmodo -admcs=*
HOST-NAME	Host name of the local computer, *NONE means that no host name has been assigned.	*ftcrei -addr=* *ftmodi -addr=*
IDENTIFICATION	Instance identification of the local openFT instance.	*ftmodo -id=*
LOCAL-SYSTEM-NAME	Name of the local system.	*ftmodo -p= -l=*
ADM-TRAP-SERVER	Name or address of the partner to which the ADM traps are sent. *NONE means that the sending of ADM traps is deactivated.	*ftmodo -atpsv=*
TRAP	The TRAP settings are output here. The possible values are ON and OFF. The row CONS indicates the console traps and the row ADM the ADM traps. The columns designate the events for which traps may be generated: SS-STATE: Change of the status of the openFT subsystem (row CONS only) FT-STATE: Change of the status of the asynchronous server PART-STATE: Change of the status of partner systems PART-UNREA: Partner systems unreachable RQ-STATE: Change of the status of request administration	*ftmodo* *-tpc=* *-atp=*

Field name	Meaning and values	Command/ option
	TRANS-SUCC Requests completed successfully TRANS-FAIL: Failed requests	
FUNCT	The settings for monitoring (MONITOR row) and tracing (TRACE row) are output in this section. The individual columns have the following meanings:	
	SWITCH: Function (monitoring or tracing) activated (ON) or deactivated (OFF)	*ftmodo* *-mon=* *-tr=*
	PARTNER-SELECTION: Selection based on the partner system's protocol type. Possible protocol types: OPENFT, FTP, FTAM. ADM (administration partner) can also be output under TRACE. ALL means that all protocol types have been selected, i.e. tracing/monitoring is possible for all partner systems. NONE means that no protocol type has been selected.	*ftmodo* *-monp=* *-trp=*
	REQUEST-SELECTION: Selection based on the request type. The following are possible: ONLY-SYNC/ONLY-ASYNC (only synchronous or only asynchronous requests) ONLY-LOCAL/ONLY-REMOTE (only locally or only remotely submitted requests). ALL means no restriction, i.e. all requests.	*ftmodo* *-monr=* *-trr=*
	OPTIONS (only in the TRACE row) NONE means no options, the trace is written normally. NO-BULK-DATA means minimum trace, i.e. bulk data (file contents) is not logged. In addition, no repetitions of data log elements are logged.	*ftmodo -tro=*

5.31 ftshwp - Display FT profiles

ftshwp stands for "show profile" and allows you to obtain information about FT profiles. In short form, it displays the names of the selected FT profiles, as well as the following information:

– whether or not the FT profile is privileged: asterisk (*) before the profile name
– whether or not the transfer admission is disabled: exclamation mark (!) before the profile name.

You can only obtain information about your own FT profiles.

Format

ftshwp -h |
[<profile name 1..8> | @s]
[-s=[<transfer admission 8..32> | @a | @n]
[,<user ID 1..32> | @a | @adm]]
[-l][-csv]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

profile name | **@s**
Is the name of the FT profile you wish to see.

@s for *profile name*
Provides information on the standard admission profile for the user ID.

profile name not specified
Profile name is not used as a criterion for selecting the FT profile to be displayed. If you do not specify the profile with -s (see below), FTAC will display information on all of your FT profiles.

-s=[transfer admission | **@a** | **@n**][**,**user ID | **@a**]
-s is used to specify criteria for selecting the FT profiles to be displayed.

If you wish to view standard admission profile, you can only specify *@n* or *@a*.

Transfer admission
: Is the transfer admission of the FT profile to be displayed. A binary transfer admission must be specified in hexadecimal format in the form x\'...\' or X\'...\' .

@a for *transfer admission*
: Displays information either on the FT profile specified with *profile name* (see above) or (if no *profile name* was specified) on all of your FT profiles.

@n for *transfer admission*
: displays information on FT profiles that do not have a defined transfer admission.

transfer admission not specified
: causes FTAC to query the transfer admission on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the transfer admission. To exclude the possibility of typing errors, the program prompts you to enter the transfer admission a second time. If you just press <ENTER>, this has the same effect as specifying *@a*.

,user ID
: must be your own login name if you are a normal user.

@a for *user ID*
: allows you to display only profiles belonging to your own login name.

@adm for *user ID*
: For the FTAC and ADM administrator only.

user ID not specified
: displays only profiles belonging to the user's own login name, regardless of who issues the command.

-s not specified
: if no profile name is specified, displays all the FT profiles belonging to the login name under which the *ftshwp* command is issued. Otherwise, displays information on the FT profile with the specified name.

-l displays the contents of the selected FT profiles.

In long form, the entire contents of the selected FT profiles are displayed. The USER-ADM parameter contains the following information:
- the login name for which an admission profile is valid or if it is an ADM profile
- whether or not it is valid for a specific password of the login name
- whether or not it is valid for any password of the login name
- whether or not it has an undefined password and is thus disabled.

USER-ADM=	**Meaning**
(user ID,,OWN)	Profile is valid for all passwords of the login name.
(user ID,,YES)	The profile is valid only for a specific password of the login name (specified in *-ua=user ID, password* with an *ftcrep* or *ftmodp* command). The profile is deactivated (not disabled) if the password is changed. You can activate it again, for example, by resetting the password.
(user ID,, NOT-SPECIFIED)	The FTAC administrator created or modified the FT profile knowing only the login name. As a result, the profile was disabled. You must enable the profile with *ftmodp* and the *-v=y* parameter.

If an FT profile is disabled, the *TRANS-ADM* parameter indicates the reasons why the profile was disabled. The following table shows the possible parameter values, as well as their meanings:

TRANS-ADM=	**Possible cause and action**
NOT-SPECIFIED	The FTAC administrator created the FT profile without transfer admission, or the FTAC user did not specify transfer admission. Measure: specify transfer admission
DUPLICATED	An attempt was made to create an FT profile with the same transfer admission. Measure: specify new transfer admission

TRANS-ADM=	Possible cause and action
LOCKED (by_adm)	The FTAC administrator modified the FT profile by login name only. The transfer admission remained unchanged but was disabled. Measure: enable the profile using the *ftmodp* command and the *-v=y* parameter
LOCKED (by_import)	The FT profile was created using the *ftimpe* command. The transfer admission remains unchanged, but is marked as disabled. Measure: enable the profile using the *ftmodp* command and the *-v=y* parameter.
LOCKED (by_user)	The FTAC user disabled his/her own FT profile. Measure: enable profile using the *ftmodp* command and the *-v=y* parameter.
EXPIRED	The time up to which the transfer admission can be used has expired. Measure: enable profile using the *ftmodp* command and the *-d* parameter, by removing the temporal restriction using the *-d* entry and defining a new time span with *-d=date*.

ftshwp does not provide a means of displaying a transfer admission. If you have forgotten a transfer admission, you have to define a new one using *ftmodp*.

-l not specified
: displays only the names of your FT profiles. Markings also indicate whether or not an FT profile is privileged (*) and whether or not it is disabled (!).

-csv You can use *-csv* to specify that the FT profiles are to be output in the CSV format. The values in the output are separated by semicolons. If *-csv* is specified, output is always in long form (analogous to *-l*) regardless of whether or not *-l* has also been specified.

-csv not specified
: The FT profiles are output in the standard format, i.e. in abbreviated form if *-l* is not specified and in detailed form if *-l* is specified.

Examples

1. Scrooge McDuck wishes to see the FT profile *goldmrep* under his login name. This profile was created in the "Example" on page 188.

   ```
   ftshwp␣goldmrep␣-l
   ```

   The output is as follows:

   ```
   goldmrep
    EXP-DATE    = 20093112
    TRANS-DIR   = FROM
    PARTNER     = goldmine
    FILE-NAME   = monthlyreport_goldmine01
    WRITE       = REPLACE-FILE
    USER-ADM    = (scrooge,,OWN)
    FT-FUNCTION = (TRANSFER-FILE, FILE-PROCESSING)
    SUCC-PROC   = 'lpr monthlyreport_goldmine01'
    FAIL-PROC   = NONE
    LAST-MODIF  = 2009-03-27 14:55:23
   ```

   The timestamp of the most recent change is shown under LAST-MODIF. If you specify *ftmodp goldmrep* without any further parameters, you can force the timestamp to be updated without changing the profile properties.

2. Scrooge McDuck wishes to see the standard FT profile:

   ```
   ftshwp @s -l
   ```

   ```
   *STD
    TRANS-ADM   = (NOT-SPECIFIED)
    WRITE       = NEW-FILE
    USER-ADM    = (scrooge,,OWN)
    FT-FUNCTION = (TRANSFER-FILE)
    LAST-MODIF  = 2009-03-22 16:06:55
   ```

5.32 ftshwptn - Display partner properties

You use the *ftshwptn* command to call up the following information about the partner systems entered in the partner list:

– The name of the partner system

– The status of the partner system (activated, deactivated)

– The security level that was assigned to the partner system

– The priority that was assigned to the partner system

– The setting for the openFT trace function for the partner system

– The number of file transfer requests to the partner system issued in the local system that have not yet been completed

– The number of file transfer requests for the local system that have been issued in the partner system

– The mode for sender verification and authentication

– The partner system's transport address, possibly with the port number if this is different from the default

– The identification of the partner system

– The routing information if the partner system can only be accessed via an intermediate instance

You can also output the partners in the partner list as a platform-specific command sequence. In this way, it is possible to save the partner list and load it at another computer which may possibly be running a different operating system.

Format

ftshwptn -h |
[<partner 1..200> | @a]
[-st=a | -st=na | -st=d | -st=ie | -st=nc | -st=ad | -st=da]
[-l | -csv | -px | -pw | -p2 | -pz]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

partner | **@a**
Specifies the partner whose properties you want to display. You can specify the name of the partner in the partner list or the address of the partner system. For details in address specifications, see section "Defining the partner computer" on page 90.

@a for *partner*
The properties of all the partners in the partner list are displayed.

partner not specified
The properties of all the partners in the partner list are displayed.

-st=a | **-st=na** | **-st=d** | **-st=ie** | **-st=nc** | **-st=ad** | **-st=da**
This operand enables you to display the properties of partner systems which have a specific status. You an specify the following values:

a (active)
All the partner systems with the status ACTIVE are displayed.

na (not active)
All the partner systems which do **not** have the status ACTIVE are displayed.

d (deactivated)
All the partner systems with the status DEACTIVE are displayed.

ie (installation error)
All the partner systems with the status LUNK, RUNK, LAUTH, RAUTH, NOKEY or IDREJ are displayed.

nc (not connected)
All the partner systems with the status NOCON or DIERR are displayed.

ad (active + automatic deactivation)
All the partner systems for which the option AUTOMATIC-DEACTIVATION is set (see the option *-ad* in the *ftaddptn* and *ftmodptn* commands) but are still active are displayed.

da (deactivated + automatic deactivation)
All the partner systems which have actually been deactivated because of the AUTOMATIC-DEACTIVATION option are displayed.

-st not specified

The output is not restricted to partner systems with a specific status.

-l | **-csv** | **-px** | **-pw** | **-p2** | **-pz**

These options determine the scope and format of the output.

-l The properties of the partner systems are output in full as a table.

-csv The properties of the partner systems are output in CSV format. The individual values are separated by semicolons.

-px The properties of the partner systems are output as a command sequence. This can be called in Unix systems as a shell procedure in order to generate partner entries with identical properties.

-pw The properties of the partner systems are output as a command sequence. This can be called in Windows systems as a batch procedure in order to generate partner entries with identical properties.

-p2 The properties of the partner systems are output as a command sequence. This can be called in BS2000 systems as an SDF procedure in order to generate partner entries with identical properties.

-pz The properties of the partner systems are output as a command sequence. This can be called in z/OS systems as a CLIST procedure in order to generate partner entries with identical properties.

-l, *-csv*, *-px*, *-pw*, *-p2*, *-pz* not specified

If you do not specify any of these options then the partners' properties are output in their abbreviated form.

Output format of ftshwptn

Examples

```
$ftshwptn
NAME     STATE SECLEV PRI  TRACE   LOC   REM P-CHK  ADDRESS
Testsys  ACT   STD    NORM FTOPT     0     0 FTOPT  D123S456.mydomain.com
tam01    ACT   5      NORM FTOPT     0     0        ftam://%ip123.11.22.33
ftamfsc  ACT   STD    NORM FTOPT     0     0        ftam://PC01.tt.net
ftamdex  ACT   STD    NORM FTOPT     0     0        ftam://PC02:102.TS1.PS1
BS2HOST  DEACT STD    LOW  FTOPT     0     0        BS2HOST
ftp001   ACT   STD    LOW  FTOPT     0     0        ftp://UX000002

ftshwptn -l
NAME     STATE SECLEV   PRI  TRACE  LOC   REM P-CHK ADDRESS
                                                   ROUTING  IDENTIFICATION
pingftam ACT   50       NORM OFF      0     0       ftam://PING.homenet.de
PINGO    ACT   STD      NORM ON       0     0 T-A   PINGPONG.homenet.de:1234
                                                            PINGPONG.homenet.de
rout0001 ACT   STD      HIGH FTOPT    0     0 FTOPT INCOGNITO
                                                   ROUT01  INCOGNITO.id.new
servftp  ACT   B-P-ATTR LOW  ON       0     0       ftp://ftp.homenet.de
```

Explanation of output

NAME
: Name of the entry in the partner list.

STATE
: Specifies how file transfer requests issued locally to the specified partner system are processed.

  ACT
  : File transfer requests issued locally to this partner system are processed with *ftstart*.

  DEACT
  : File transfer requests issued locally to this partner system are initially not processed, but are only placed in the request queue.

  ADEAC
  : Failed attempts at establishing a connection lead to this partner system being deactivated. The maximum number of consecutive failed attempts is 5. In order to perform file transfers with this partner system again, it must be explicitly reactivated with *ftmodptn -st=a*.

  NOCON
  : Attempt to establish a transport connection failed.

  LUNK
  : Local system is not known in the remote FT system.

RUNK
: Partner system is not known in the local transport system.

AINAC
: Partner system has been deactivated after a number of unsuccessful attempts to establish a connection.

LAUTH
: Local system could not be authenticated in the partner system. A valid public key for the local openFT instance must be made available to the partner system.

RAUTH
: Partner system could not be authenticated in the local system. A valid public key for the partner system must be stored in the folder *syskey* of the openFT instance. In the case of the default instance, *syskey* is in the directory */var/openFT/std*.

DIERR
: A data integrity error has been detected on the connection to the partner system. This can be the result of attempts at manipulation on the data transfer path or of an error in the transport system. The connection has been interrupted, but the affected request is still live (if it has the capability of being restarted).

NOKEY
: The partner does not accept unencrypted connections, but no key is available in the local system. A new key must be generated.

IDREJ
: The partner or an intermediate instance has not accepted the instance ID sent by the local system. Check whether the local instance ID matches the entry for the partner in the partner list.

SHORT
: A resource bottleneck has occurred on the partner.

SECLEV
: Security level assigned to the partner system.

1..100
: A fixed security level is assigned to the partner system:
1 is the lowest security level (partner is extremely trusted) and 100 is the highest security level (partner is not trusted).

STD
: The global setting for the security level applies.

B-P-ATTR
: The security level is assigned to the partner on the basis of the partner's attributes, i.e.:
  – Security level 10 if the partner has been authenticated.
  – Security level 90 if the partner is known in the transport system and is identified by the name it is known by in the transport system.
  – Security level 100 otherwise, i.e. if the partner has only been identified by its address.

PRI Priority of a partner with respect to the processing of requests:

NORM
: Normal priority.

LOW Low priority.

HIGH High priority.

TRACE
: The global settings for partner selection in the openFT trace function apply.

FTOPT
: The global setting for partner selection in the openFT trace function applies.

ON The trace function is activated for this partner. However, a trace is only written if the global openFT trace function is also activated.

OFF The trace function is deactivated for this partner.

LOC Shows the number of file transfer requests addressed to the partner system entered in the local system.

REM Shows the number of file transfer requests issued by the remote FT system and addressed to the local FT system.

P-CHK
: Shows the settings for sender verification and authentication.

FTOPT
: The global setting for sender verification applies.

STD Checking of the transport address is deactivated. Only the identification of the partner is checked. The transport address of the partner is not checked even if extended sender verification is activated globally.

T-A Checking of the transport address is activated. The transport address of the partner is checked even if checking of the transport address is deactivated globally. If the transport address used by the partner to log in does not correspond to the entry in the partner list, the request is rejected.

AUTHM
Authentication is activated.

NOKEY
No valid key is available from the partner system although authentication is required.

ADDRESS
Address of the partner system.

ROUTING
Routing info of the partner system if specified. The routing info is only output with *ftshwptn* -l.

IDENTIFICATION
Identification of the partner system if specified. The identification is only output with *ftshwptn -l*.

5.33 ftshwr - Display request properties and status

The *ftshwr* ("show requests") command allows you to request information about FT requests. You can specify selection criteria in order to obtain information about specific FT requests.

Users can only obtain information about the requests they own.

Format

ftshwr -h |
[-ua=<user ID 1..32> | -ua=@a]
[-ini=l | -ini=r | -ini=lr | -ini=rl]
[-st=a | -st=w | -st=l | -st=c | -st=f | -st=h | st=s]
[-pn=<partner 1..200>]
[-fn=<file name 1..512>]
[-s | -l][-csv]
[<request ID 1..2147483647>]

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

-ua=user ID | **-ua=@a**
You use *-ua* to specify the user ID for which requests are to be displayed.

user ID
As a user, you can only specify your own user ID.
As an FT administrator, you may specify any user ID here.

@a As an FT administrator, you can specify @a to display requests for all user IDs.

-ua= not specified
Your own user ID is the selection criterion.
Exception: The FT administrator has called the command and also specified a request ID: in this case, the presetting is *@a*.

-ini=l | **-ini=r** | **-ini=lr** | **-ini=rl**

You use *-ini* to specify the initiator for which you want to display requests. The following specifications are possible:

l (local) Only locally submitted requests are displayed.

r (remote) Only remotely submitted requests are displayed.

lr, rl (local + remote) Both locally and remotely submitted requests are displayed.

-ini not specified

The initiator is not the selection criterion (corresponds to *lr* or *rl*).

-st=a | **-st=w** | **-st=l** | **-st=c** | **-st=f** | **-st=h** | **-st=s**

If you specify *-st* then only information on requests with the corresponding status is output.

The following specifications are possible:

a (active)
The request is currently being executed.

w (wait)
The request is waiting to be executed.

l (locked)
The request is locked.

c (cancelled)
The request has been deleted.

f (finished)
The request has already been executed.

h (hold)
The starting time specified on the issue of the request has not yet been reached.

s (suspend)
The request was interrupted, i.e. it is currently in the SUSPEND status.

-pn=partner
: You use *-pn* to specify a name or an address for the partner system for which you want to display requests. The partner should be specified as on request submission or as output by the *ftshwr* command without the *-s*, *-l* or *-csv* option. If openFT finds a partner in the partner list for a specified partner address then *ftshwr* displays the name of the partner even if a partner address was specified at the time the request was entered.

-fn=file name
: You use *-fn* to specify the file name for which requests are to be displayed. Requests that access this file in the local system are displayed.

  You must specify the file name that was used when the request was issued. This file name is also output by the *ftshwr* command without the *-fn* option.

  Wildcards are not permitted in the file name.

-s (sum) specifies that a summary overview of requests is to be output. For each possible request status (see the *-st* option), this overview indicates the number of requests that have this status.

-l (long form) specifies that the request properties are to be output in full.

-csv Specifies that the request properties are to be output in CSV format. If you also specify *-s* then the summary overview is output in CSV format. The values in the overview are output separated by semicolons.

-s, *-l* and *-csv* not specified
: The request attributes are output in standard form.

request ID
: *request ID* specifies the identification of a specific request that is to be output. The request ID is output on the screen on acknowledgment of receipt of the request. It can also be viewed, for example, using the *ftshwr -l* command.

If you have specified a request ID and the other specified criteria do not correspond to the request then the request is not displayed and the following error message is output:

`ftshwr: Request` ***request ID*** `not found`

5.33.1 Output of the ftshwr command

5.33.1.1 Standard ftshwr output

```
$ftshwr
TRANS-ID   INI STATE PARTNER  DIR  BYTE-COUNT  FILE-NAME
65558      LOC WAIT  *PINGO   TO   0           /home1/september.pdf
196610     LOC WAIT  servus.* FROM 0           /home2/mails/memo02.txt
262146     LOC WAIT  servus.* TO   0           /home3/pic/picture10.gif
```

Description of the output

TRANS-ID
: The TRANS-ID column (transfer identification) contains the request numbers used by openFT to identify the file transfer requests. The TRANS-ID can be used to cancel requests with the *ftcanr* command.

INI
: The INI column indicates the initiator:
LOC: The request was submitted in the local system.
REM: The request was submitted in the remote system.

STATE
: The STATE column indicates the priority of the request.

  The priority is displayed after the state identifier. The only possible display is *l* for "low". If the request has the priority *normal* then nothing is displayed.

  The following states are possible:

  ACT (active)
  : The request is currently being processed.

  WAIT (wait)
  : The request is waiting.
  In this case, the partner system (PARTNER) may be indicated.
  This indication shows the cause of the *WAIT* state.

  LOCK (locked)
  : The request is temporarily excluded from processing.
  This state may occur both for openFT and for FTAM partners.
  With openFT partners, e.g. when a resource bottleneck is encountered or when external data media must be made available.
  With FTAM partners, when one of the partners proposes a waiting period until the next start or recovery attempt via the FTAM protocol, and this period exceeds the delay normally permitted.

In this case, the partner system (PARTNER) may be indicated. This indication shows the cause of the *LOCKED* state.

CANC (canceled)
: The request was cancelled in the local system.
However, the remote system is aware of its existence, e.g. because it was previously active. Therefore, the request cannot be removed from the request queue until a connection to the partner has been re-established.

FIN (finished)
: This status arises for requests involving FTAM partners when the request has been either completed or cancelled, but the user has not yet been informed of the fact.

HOLD (hold)
: The start time specified when the request was issued has not been reached.

SUSP (suspend)
: The request was interrupted.

PARTNER
: Name or address of the partner, see also page 90. If the partner address is more than 8 characters in length then it is truncated to 7 characters and suffixed with an asterisk (*).

If the request is in a WAIT or LOCKED state, the following indicators before PARTNER are also entered in the request queue:

␣ (empty) No resources free at present (e.g. no memory).

* The local FT administrator has locked the resource, e.g. deactivating the partner.

! Connection setup to the partner system failed. The partner is currently inactive, or it can currently accept no further connections, or a network node has crashed.
Other possibilities: The connection to the partner system has been lost; a data integrity error has been detected.

? An installation or configuration error has occurred (e.g. the local system is not known to the partner), authentication of one of the partners has failed, or the encryption is local, or not available to the partner system.

DIR The DIR column specifies the direction of transfer.

TO Send to the remote system.

FROM
Fetch from the remote system.

BYTE-COUNT
This column indicates the number of bytes transferred and saved up to now. The BYTE-COUNT counter is only updated at certain intervals.

FILE-NAME
Name of the file in the local system.

5.33.1.2 Totaled ftshwr output

In the case of totaled output, a table showing the number of requests in the various request states is output (refer to the *State* column under the standard output for the meanings of the states):

```
ftshwr -s
 ACT   WAIT   LOCK   SUSP   HOLD    FIN      TOTAL
   3      2      0      0      0      0          5
```

5.33.1.3 Detailed output from ftshwr

Example for the detailed output of the request with request ID 131074:

```
ftshwr -l 131074

TRANSFER-ID =131074      STORE  =09-05-29 11:45:27  FILESIZE=514610
   STATE    =WAIT        BYTECNT=0
   INITIATOR=LOCAL       TRANS  =FROM               PRIO    =NORM
   WRITE    =REPLACE     START  =SOON               CANCEL  =NO
   COMPRESS =NONE        DATA   =CHAR
   TRANSP   =NO          ENCRYPT=NO
   OWNER    =maier       DICHECK=NO                 RECFORM =VARIABLE
   PARTNER  =ftserv01.mycompany.net
   PARTNER-STATE = ACT
   PARTNER-PRIO  = NORM
   LOC: FILE     =/home2/memo02.txt
        TRANS-ADM=(maier)
        CCSN     =ISO8859-1
   REM: FILE     =/home/save/memo02.txt
        TRANS-ADM=(servelog)
```

Description of the output

TRANSFER-ID (transfer identification)
: Request ID which openFT uses to identify file transfer requests. Requests can be canceled using the *ftcanr* and the request ID.

STATE
: State of the request. Possible values:

  ACTIVE
  : The request is currently being processed.

  WAIT
  : The request is waiting. If the cause of the WAIT state is known, further information is indicated in the PARTNER-STATE field.

  LOCKED
  : The request is temporarily excluded from processing. This status can also occur at openFT and at FTAM partners.
  With openFT partners, when a resource bottleneck is encountered or if external data media must first be made available for example.
  With FTAM partners, when one of the partners proposes a waiting period until the next start or recovery attempt via the FTAM protocol, and this period exceeds the delay normally permitted.
  If the cause of the LOCKED state is known, further information is indicated in the PARTNER-STATE field.

  CANCELLED
  : The request was cancelled in the local system. However, the remote system is aware of its existence because, for example, it was previously active. Therefore, the request cannot be removed from the request queue until the connection to the partner has been re-established.

  FINISHED
  : This status occurs for requests involving FTAM partners when the request has either been completed or cancelled, but the user has not yet been informed of this.

  HOLD
  : The start time specified when the request was issued has not yet been reached.

  SUSPENDED
  : The request was interrupted.

INITIATOR
: This specifies where the request was issued. Possible values:

  LOCAL
  : The request was issued in the local system.

  REMOTE
  : The request was issued in the remote system.

WRITE
: This specifies whether the destination file is to be overwritten, extended or created. Possible values:

  OVERWRITE (default value)
  : If the destination file already exists, it is overwritten; otherwise, it is created.

  EXTEND
  : If the destination file already exists, the file sent is appended to the destination file; otherwise. If the destination file did not exist, it is created.

  NEW
  : A new destination file is created and written.

COMPRESS
: This specifies whether the file should be transferred with data compression.

  Possible values: BYTE, ZIP, NONE.

TRANSP
: Indicated whether the file is to be sent in transparent file format. Possible values: YES, NO

OWNER
: Local login name.

PARTNER
: Name or address of the partner, see also page 90.

PARTNER-STATE
: Status of the partner. Possible values:

  ACT Activated

  DEACT
  : Deactivated

NOCON
: No connection, for example because the openFT server has not been started in the remote system.

INSTERR
: An installation or configuration error has occurred (the local system is not known to the partner, for instance), authentication of one of the partners has failed, or the encryption is local, or not available to the partner system.

SHORT
: A resource bottleneck has occurred on the partner.

PARTNER-PRIO
: Prioritization of the partner when processing requests.
Possible values:

LOW The partner has low priority.

NORM
: The partner has normal priority.

HIGH
: The partner has high priority.

LOC Properties in the local system:

FILE File name in the local system

TRANS-ADM
: Transfer admission for the local system

CCSN
: CCS name used in the local system. The CCSN is only output for text files.

SUCC-PROC
: Local follow-up processing commands if successful
(if specified in the request)

FAIL-PROC
: Local follow-up processing commands if unsuccessful
(if specified in the request)

REM Properties in the remote system:

FILE File name in the remote system

TRANS-ADM
Transfer admission in the remote system. Possible values:

REMOTE-PROFILE
request with FTAC transfer admission

TRANS-ADM=(*user ID*)
request with *user ID,,password*

CCSN
CCS name used in the remote system

SUCC-PROC
Remote follow-up processing commands if successful (if specified in the request)

FAIL-PROC
Remote follow-up processing commands if unsuccessful (if specified in the request)

STORE
Indicates the time at which the request was entered in the request queue.

BYTECNT
This value is output only if the request is currently active or if it was already active and the file transfer has been interrupted. BYTECNT indicates the number of bytes transferred and saved up to now. The counter is updated regularly.

TRANS
This shows the direction of transfer. Possible values are:

TO The document is sent.

FROM The document is received.

START
Indicates the time at which the request is to be started. Possible values:

Date / Time
The date and time at which the request is to be started is output.

SOON
The request should be started as soon as possible.

No entry
The request was issued in the remote system.

DATA
Indicates the file type. Possible values:

CHAR (default value for openFT partners)
The file contains text with variable record lengths.

BIN The file contains an unstructured sequence of binary data.

USER
The file contains structured binary data with variable record length.

ENCRYPT
Indicates whether data encryption was specified.
Possible values: NO, YES.

DICHECK
Specifies whether the integrity of the data is to be checked.
Possible values: NO, YES.

FILESIZE
Size of the file in bytes. If the output is followed by a "K", the output is in kilobytes. If it is followed by an "M", the output is in megabytes. The size is indicated here only if the request was already active. For receive requests, a value is indicated here only if the partner has sent one with the request.

PRIO Request priority. Possible values:

NORM
The request has normal priority

LOW
The request has low priority

No entry
The request was issued in the remote system.

CANCEL
If the "Cancel-Timer" was set, the time at which the request is deleted from the request queue is indicated here. If no cancel time was specified in the request, NO is output.

RECFORM
: Record format.
Possible values: UNDEFINED, VARIABLE, FIX.

RECSIZE
: Maximum record size, if specified.

DIAGCODE
: This column is usually empty. Otherwise, it provides further diagnostic information on operational states in the form of a CMX return code or an FTAM or openFT diagnostic code. FTNEA diagnostic codes have the format NEBFnnnn (NEABF) or NEBDnnnn (NEABD). The following openFT diagnostic codes have been defined:

Value	Meaning
0	No cause specified.
1	Connection setup normal.
2	There is a resource bottleneck.
3	There is a resource bottleneck; the connection will be set up later by the rejecting entity.
4	Initialization is not yet complete.
5	SHUTDOWN is in progress.
6	The requesting entity is unknown.
7	A protocol error has occurred.
8	A transport error has occurred.
9	A system error has occurred.
10	This code is reserved (for SN77309 part 5).
11	The connection is not accepted without encryption.

FTAM diagnostic codes have the format `FTAMnnnn` and values from the ISO 8571-3 standard. An extract of possible diagnostic codes taken from the standard can be found in the section “FTAM diagnostic codes as per ISO 8571-3” on page 123.

The following values are only output for FTAM partners:

STOR-ACCOUNT
: Account number; is output only if specified by the user.

AVAILABILITY
: Possible values: IMMEDIATE, DEFERRED.
Is output only if specified by the user.

ACCESS-RIGHTS
: Access mode

Possible values: combinations of r, i, p, x, e, a, c, d.
Is output only if specified by the user.

LEGAL-QUAL
: Legal qualification

Is output only if the local system is the initiator and the value is specified by the user.

5.34 ncopy - Synchronous file transfer

The *ncopy* command is used to issue synchronous requests for sending one or several files to a remote system or for fetching a file from a remote system or for executing an operating system command in the local or remote system. The *ncopy* command is executed even if the asynchronous openFT server has not been started.

Instead of a local file, you can also use standard input (*stdin*) when sending a file, and standard output (*stdout*) when receiving a file.

If openFT rejects your request, an error message will be displayed explaining why it was rejected (see chapter “Messages” on page 377).

openFT transfers the file synchronously to the user process or executes the remote command.

Only one file can be fetched from a remote system for each *ncopy* command. If you want to fetch several files synchronously, use the *ft_mget* command. See the section “ft_mget - Fetching multiple files” on page 423.

Status message

openFT displays a status message while file transfer is in progress. The syntax of this message is as follows:

*b*KB [*p*%; [*hh*:]*mm*:*ss*]

The variables are:

b Number of bytes (in KB) already transferred

p Percentage of file already transferred

hh:mm:ss
estimated time to completion of transfer in hours, minutes and seconds. The hours are not displayed unless the time to completion is longer than sixty minutes. If the size of a file for a receive request is unknown, only the counter for the number of bytes transferred is active.

The status message is updated every three seconds. The first message does not include the anticipated time to completion of transfer.
You receive status information only if

– the file is correspondingly large,
– the *-S* or *-s* switch was not set to suppress messages,
– the request is not running as a background process (*ncopy .... &*),
– the standard error output (*stderr*) is not redirected to a file,
– a file was specified as source file or the data was input via a pipe (dash (-) for source file), i.e. not input via keyboard.

If the size of the send file is unknown, the status message merely shows the number of bytes already transferred. This is the case if the data is input via a pipe or when a file is received.

When the transfer has been successfully completed, openFT outputs a result message on the screen (*stderr*) of the user with the following format:

`ncopy: request` *request ID.* `File` *file name* `transferred`

If openFT was not able to execute your request successfully, an error message will be displayed on the screen (see chapter “Messages” on page 377).

A number of special considerations apply for transfer requests with FTP partners. See the section “FTP partners” on page 33.

Format

```
ncopy  -h  |
       [ -t | -u | -b ][ -x ]
       [ -o | -e | -n ]
       [ -k | -z ][ -c ][ -S | -s ] [ -m=n | -m=f | -m=a ]
       [ <file name 1..512>   [<file name 1..512>...<file name 1..512>] | -
           <partner 1..200>![<file name 1..512> | <prefix 0..511>%] ] |
       [ <partner 1..200>![<file name 1..512>]
           <file name 1..512> | <prefix 0..511>% | - ]
       [ <transfer admission 8..67> | @n | @d |
         <user ID 1..67> [,[<account 1..64>] [,[<password 1..64>]]] ]
       [ -p=[<password 1..64>] ][ -di ]
       [ -lc=<CCS name 1..8> ][ -rc=<CCS name 1..8> ]
       [ -rs=<follow-up processing 1..1000> ]
       [ -rf=<follow-up processing 1..1000> ]
       [ -r=v[<1..32767>] | -r=f[<1..32767>] | -r=u[<1..32767>] |
         -r=<1..32767> ]
       [ -tff=b | -tff=s ][ -trf=u ]
       [ -tb=n | -tb=f | -tb=a ]
       [ -av=i | -av=d ] [ -ac=<new account 1..64> ]
       [ -am=[r][i][p][x][e][a][c][d] | -am=@rw | -am=@ro ]
       [ -lq=<legal qualification 1..80> ]
       [ -cp=[<password 1..64>] ]
```

Description

-h Displays the command syntax on the screen. Entries after the *-h* are ignored.

[**-t** | **-u** | **-b**][**-x**]
Identifies the type of file in the local system.

If you send a file to an FTAM partner without specifying a file type, the file type is determined by the structure entries of the send file. The structure entries can be displayed by outputting the local openFT attributes (*ftshwf file name -l*). If there are no structure entries, the default value is *-t*. If you fetch a file from an FTAM partner without specifying a file type, the file type is determined by the file attributes in the FTAM partner. For more detailed information about file types when dealing with FTAM partners, see the section “Mapping FTAM attributes to the real file system” on page 114.

-t (default value with openFT partners)
The file contains text with variable-length records. Records end with the linefeed character \n.

Maximum record length = 32767 bytes.

-u The file contains binary data with variable record length structured by the user. Each record starts with 2 bytes which contain the length data for the record.

Maximum record length = 32767 bytes

-b The file contains user-structured binary data with variable-length records.
For further information, see "Binary transfer" on page 81.

-x The send file is transferred in a transparent file format and is stored in the destination system, i.e. this is a file whose attributes are transparent for the local system. The local system here acts as a storage and/or transport medium.

If a file is transparently retrieved with *-x* for local buffering, then it must be sent again to the remote system in binary form (i.e. with *-b*).

-o | **-e** | **-n**
Indicates whether the destination file is to be newly created, overwritten, or extended.

-o (default value)
The destination file will be overwritten. A new destination file will be created if it did not already exist.

-e The transferred file will be appended to an existing destination file. A new destination file will be created, if it did not exist already.

-n The destination file will be newly created and written. If the destination file already exists, the request will be rejected. In this way, you can protect a file from being overwritten inadvertently.

-k Indicates that identical characters repeated consecutively are to be transferred in compressed form (byte compression). In the case of connections to partners which do not support this type of compression, no compression are used automatically.

-z Indicates that zip compression is used. In the case of connections to partners which do not support this type of compression, byte compression (corresponds to the option *-k*) or no compression are used automatically.

-c Indicates that the transfer data are encrypted during file transfer. Encryption of the request description data (see page 57) is not affected by this option. If the partner system does not support data encryption, the request is rejected.

[**-S** | **-s**]
Suppresses file transfer messages to *stderr*.

-S All messages are suppressed.

-s The status message and the end messages are suppressed; error messages are output.

-m=n | **-m=f** | **-m=a**
This indicates whether the result message is to be deposited in the mail box of the user who issued the request.

n (default value)
The result message is not deposited in the mailbox.

f The result message is only deposited in the mailbox in the event of errors.

a The result message is always deposited in the mailbox.

file name1 [file name2.. [file name]] | **-** partner**!**[file name | [prefix]**%**] | partner**!**[file name] file name | **-** | [prefix]**%**
specifies the source and destination. The syntax depends on the direction of transfer selected and if pre- or postprocessing commands are used.

Sending without pre/postprocessing

Source	**Destination**
local file1 [*local* file2 ..]	**-**	partner**!**[*remote* file	[prefix]**%**]

Fetching without pre/postprocessing

Source	**Destination**
partner**!**[*remote* file]	*local* file	**-**	[prefix]**%**

Sending and fetching with pre- or postprocessing

If you want to perform pre- or postprocessing, then you must enter an operating system command instead of the local or remote file name (in the syntax of the corresponding system):

Sending with preprocessing

Source	Destination
"**!***local* command"	Partner**!**[*remote* file]

Sending with postprocessing

Source	Destination
local file1 [*local* file2 ..] **|** **-**	Partner**!"!***remote* command"

Fetching with preprocessing

Source	Destination
Partner**!"!***remote* command"	*local* file **|** **-**

Fetching with postprocessing

Source	Destination
Partner**!**[*remote* file]	"**!***local* command"

You can also combine preprocessing and postprocessing in the same request.

A maximum of 712 characters may be specified for *source* and *destination* (maximum 512 characters for the file name and maximum 200 for the partner). Please note that the maximum lengths of file names are system-dependent; for example, in Unix systems it is 512 and in Windows systems a maximum of 256 characters.

local file1 [local file2 ..]

Sending: The name(s) of the local file(s) have to be entered here. If you send several files, you have either to specify %, %BASENAME or %FILENAME for the remote file name, see

below, or you specify one remote file name and use option *-e*. With *-e*, the transferred files are concatenated and written in the specified remote file.

The specification of UNC names is also possible.

Wildcards
If you wish to send several files to a remote system and the files should have the same names in the remote system, you may use wildcards. Do this using the asterisk (*) commonly used for example. The file name must not contain exclamation marks (!). If you specify commands for follow-up processing, follow-up processing is carried out for each file.

Fetching: Enter the name of the receive file.
The local file name may be an absolute or relative path name.
If the file name ends with %unique or %UNIQUE, this string is replaced by a string which changes for each new call. In addition, a suffix separated by a dot may be specified after %unique or %UNIQUE, e.g. `file1%unique.txt`.
However, *ncopy* will not create a directory that does not already exist.

- (dash) for *local file*

Sending: The dash for local file stands for standard input *stdin*. You can use the dash to link a Unix command with *ncopy*, for example (see example 6 for more details). You can also enter data directly via keyboard, in which case you send the *ncopy* command with a dash for the local file, before processing to enter data. Terminate your direct entry by pressing <END> or CTRL+D. See example 7 for more details.

Fetching: The dash stands for standard output *stdout*. The dash directs output to the screen. You can use the dash if you want to link the *ncopy* output with a command on the Unix system, for example.

[prefix]**%** for *local file*

Fetching: For the receive file name, you may specify *%*, *%BASENAME*, *%FILENAME* or, in addition, a prefix. These variables are substituted as follows:

% and **%BASENAME**
: are substituted by the last part of the name of the remote file. The last part of the name starts after the slash (/) or backslash (\), or a corresponding character in the remote system.

%FILENAME
: is overwritten by the full name of the remote file specified in the command.

prefix
: You may also specify a prefix for the local file name, e.g. *save.%FILENAME*. This prefix must end with a dot (.), a slash (/) or a backslash (\).

remote file
: *remote file* can be either absolute or relative to the remote transfer admission (when sending or fetching). If the file name in the remote system has been predefined in an authorization profile, it must not be specified here.

  If the file name contains blanks, they must be enclosed in double quotes (e.g. "file name").

  If the partner system is running openFT for BS2000/OSD, elements from PLAM libraries may also be specified here (syntax: Libname/Element type/Element name).

  If the file name ends with %unique or %UNIQUE, this string is replaced by a string which changes for each new call. In addition, a suffix separated by a dot may be specified after %unique or %UNIQUE if the partner is a Unix or Windows system.

  If the file name of a receive request starts with an pipe character ("|"), the file name is executed on the remote system as a command if the remote system supports the preprocessing function.

[prefix]**%** for *remote file*
: **Sending:** If you are sending several files, you have to specify *%*, *%BASENAME*, *%FILENAME* for the remote file name. In addition, you can specify a prefix. These variables are substituted as follows:

% and **%BASENAME**
: are substituted by the last part of the name of the send file. The last part of the name starts after the slash (/) or backslash (\), or a corresponding character in the send system.

  Please note that when you use % and %BASENAME with wildcards, files with the same names can be produced during substitution and that these are mutually overwritten.

  Example

```
ncopy␣file/test1.c␣test/test1.c\
␣partner!destination/% transadm
```

  Both files are copied to *destination/test1.c*.

%FILENAME
: is overwritten by the full name of the send file specified in the command.

prefix
: You may also specify a prefix for the remote file name. This name must end with a dot (.), a slash (/) or a backslash (\).

  Example

```
ncopy␣*.c␣*.txt␣test␣partner!prob.%␣profile01
```

  All files which end with *.c* and *.txt* and the *test* file are transferred to the remote system and stored there under the name *prob.*<*local filename*>. Here, *profile01* is the transfer admission.

I*command* for *file name*
: *command* is any command on the local or remote system. The "|" character (vertical bar or pipe character) must always be placed before the command. The "|" character must always be escaped by either a backslash (\) or double quotes ("), i.e. "I*command*" should always be enclosed in double quotes.

  In the case of preprocessing openFT transfers the data output by the command to standard output as a file.

  In the case of postprocessing openFT reads the transferred data from the standards input.

In the case of preprocessing, you can also pass the data to the %TEMPFILE variable and, in the case of postprocessing, read the data from the %TEMPFILE variable, see section “Preprocessing and postprocessing” on page 100.

If command execution takes longer than ten minutes, a timeout occurs on partners using openFT prior to V8.1 and command execution is regarded as having failed. On partners using openFT V8.1 and later, this restriction no longer applies.

The remote command processing in Unix or Windows systems is starting in the $HOME or Home directory of the user.

The PATH variable is used as follows in the search path for preprocessing and postprocessing commands in Unix systems:

– Default instance:
:/opt/openFT/bin:/bin:/usr/bin:/opt/bin

– Other instance:
:/var/openFT/instance/openFT/bin:/bin:/usr/bin:/opt/bin
where *instance* is the name of the relevant instance.

This means that the system first searches in the current directory (first ":"). Before calling a "real" preprocessing or postprocessing command you can switch to another directory as follows:

cd *path-name*;*command*

path-name is then used as the current directory. There must not be a blank between the semicolon and the command.

partner

partner is the name of the partner system in the partner list or the address of the partner system. For details on address specifications, see section “Defining the partner computer” on page 90.

transfer admission | **@n** | **@d** |
user ID [**,**[account][**,**password]]]

In order to be able to send a file to a remote system or to fetch one from it, you must furnish the remote system with proof of identity. For this purpose, you will need login admission in the syntax valid for the remote system. You can specify this transfer admission

– as an FTAC transfer admission if FTAC is used in the remote system,

– or as a login/LOGON admission in the syntax used by the remote system (user ID, possibly together with account or password).

For details, see section “Transfer admission” on page 94.

@n for *transfer admission*
By entering *@n* you specify that the remote system requires no login admission.

@d for *transfer admission*
Specifying *@d* (blanked transfer admission) causes openFT to query the transfer admission on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the transfer admission.

A binary password or binary transfer admission must be specified in hexadecimal format in the form x'...' or X'...'. If you enter the password directly, remember to insert a backslash (\) to escape the single quotes if you did not enclose the remote login admission in double quotes, for example: X\'c6d9e4c5\'.

password not specified
Omitting the password necessary for admission causes openFT to query the password on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the password. In this case, single quotes must not be escaped by a backslash (\).

Nevertheless, you have to specify the commas, e.g.:

```
ncopy file partner!file user-id,,
```
or
```
ncopy file partner!file user-id,account,
```

neither *transfer admission* nor *user ID* specified
causes the same as *@d*, i.e. openFT queries the transfer admission on the screen after the command is entered. Your (blanked) entry is always interpreted as transfer admission and not as user ID.

-p=[password]
If the file in the remote system is protected by a write password, you must enter this password when sending a file.
If the file is protected by a read password, then this password must be specified when fetching a file from the remote system.

A binary password must be entered in hexadecimal form x\'...\' or X\'...\'. This is of relevance for links to openFT for BS2000/OSD, because BS2000 supports the definition of hexadecimal passwords. If you enter the password directly, remember to insert a backslash (\) to escape the single quotes, for example: X\'c6d9e4c5\'.

password not specified

Specifying *-p=* causes openFT to query the write or read password on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the password. In this case, single quotes must not be escaped by a backslash (\).

-di is specified, if the data integrity of the transferred file is to be checked by cryptographic means. With it, harmful data manipulations on the transmission network are identified. In case of an error openFT performs an error recovery for asynchronous transfer requests.

If the partner system does not support the check of data integrity (e.g. openFT < V8.1), the request is denied.

For requests with data encryption (option *-c*), data integrity is automatically checked. Testing mechanisms of the transfer protocols in use automatically identify transfer errors in the network. For this purpose you do not have to specify the *-di* option.

-lc=CCS name

(local coding) specifies the type of coding (character set) to be used to read or write the local file. *CCS name* must be known in the local system.

The default value is the character set defined by the FT administrator.

Details about the CCS name and the associated code tables can be found in section “Code tables and coded character sets (CCS)” on page 85.

-rc=CCS name

(remote coding) specifies the type of coding to be used to read or write the remote file. *CCS name* must be known in the remote system.

The default value is the character set defined in the remote system by means of XHCS (BS2000/OSD) or the openFT operating parameters.

The option *-rc* is supported only by the openFT protocol and partners with openFT V10.0 or higher. Please note that not all partner systems support all the character sets that are possible in the local system. For details on CCS names and the associated code tables, see section “Code tables and coded character sets (CCS)” on page 85.

-rs=’follow-up processing’
Here you can specify a command in the syntax of the remote system. Following a **successful transfer** operation, this command is executed in the remote system under the specified login.
Further information is given in the section “Commands for follow-up processing” below.

-rf=’follow-up processing’
Here you can specify a command in the syntax of the remote system. This command will be executed in the remote system under the specified login if a **transfer** operation that has already started is **cancelled**.
Further information is given in the section “Commands for follow-up processing” below.

Commands for follow-up processing

- Entries for local follow-up processing, i.e. for *ls* and *lf*, are not possible for the *ncopy* command. The total number of characters for remote follow-up processing, i.e. for *rs* and *rf*, may not exceed 1000 characters, but this maximum value may be lower if a FT version < V10 is used in the remote system.
- The entries for follow-up processing must be enclosed in single or double quotes (’ or “).
  If the entry for follow-up processing also contains single quotes (’), it is recommended to enclose the entire entry in double quotes (“).
  The single quotes in the follow-up processing command (e.g. single quotes in a BS2000 password) can then be written as expected in the partner system (e.g. BS2000).
- When starting follow-up processing in the remote system, the specified variables are first substituted, and the follow-up processing commands are then executed. The following variables are permitted:

%FILENAME
File name in the relevant system. The entry is automatically taken from the command. If you specified the variable *%UNIQUE* (or *%unique*) for the remote file name during transfer, the *%FILENAME* variable contains the already converted (i.e. unique) file name.

%PARTNER
Name or address of the partner system in long form, i.e. with dynamic partners, all address components are taken (protocol prefix, port number, selectors, ...). *%PARTNER* is substituted by the name of the initiator system (with the name as known in the partner system).

%PARTNERAT
Name or address of the partner system in short form, i.e. with dynamic partners, only the *host* address component is taken, see page 143. In addition, each character is replaced by a '@' if it is neither a letter nor a digit or a period.

%RESULT
Message number of the request, as required by the system concerned.
If, for example, a send request is successfully executed, the value of *%RESULT* in the local system contains the message number 0 (in openFT V10 and higher).

If the partner is an openFT for BS2000/OSD system, you may also use the variables *%ELEMNAME*, *%ELEMVERS* and *%ELEMTYP*.

- Special considerations with follow-up processing in remote Windows systems
  - Only the system environment variables are available, not the user variables. In addition, the user-specific Registry entries are not loaded before follow-up processing is executed.
  - Any program can be started, e.g. a shell command, a program (*.exe* or *.com*) or a batch procedure (*.bat* or *.cmd*). If the command requires a path specification, then use the absolute path.
  - Before calling the follow-up processing, it is also possible to switch to another directory as follows:

    `cd` *path-name*`;`*command*

    path-name is then used as the current directory. There must not be a blank between the semicolon and the command. *path-name* must not be a directory which is addressed using a UNC name.

  - If you wish to execute shell-internal Windows commands (e.g. *move* or *copy*), remember that you must specify the command processor *cmd.exe /c* at the start of the command.

- Follow-up processing in the remote Unix system does not involve execution of the sequence of commands stored in the *.profile* file. Only the default values of the $HOME, $LOGNAME, $PATH, and $USER shell variables are available, as well as the values of the $LANG and $TZ variables set by *root*.
- The search path (PATH variable) for follow-up processing commands is preceded by the component */var/openFT/instance/openFT/bin*, where *instance* means the name of the corresponding instance.
- With requests for FTAM and FTP partners, the follow-up processing function is not available in the remote system (exception: -rs='∗DELETE' for FTAM receive requests to delete the send file after successful processing). If FTAC is used in the remote system, this restriction can be avoided by creating an FT profile in the remote system and defining follow-up processing for it.
- When specifying BS2000 commands, remember to insert a slash (/) at the beginning of the command.

-r=v[record length] | **-r=f**[record length] | **-r=u**[record length] | **-r=**record length
: indicates the record format and the record length. This also enables records that are longer than the default value to be transferred. However, you must bear in mind that not every record length can be processed in all partner systems.

  If you have selected file type *b* (binary), *record length* is the value for all records of the send file.

  Maximum value: 32767 bytes.

  With FTAM partners, the maximum record length specification is not valid unless the file type is set explicitly to *t*, *u* or *b*.

  It is also possible to specify the record format, see page 227:

  v variable record length, *record length* determines the maximum value

  f fixed record length, *record length* then applies to all records

  u undefined record length

  The combinations `-u -r=f`*recordlength* and `-u -r=u`*recordlength* are not permitted.

If *-r* is omitted then the following default values apply for the record format:

Option	Default value	Corresponds to
-b	u (undefined)	-r=u...
-t	v (variable)	-r=v...
-u	v (variable)	-r=v...

-tff=b | **-tff=s**
Specifies the format of the destination file.

b The destination file is to be saved as a block-structured file. This means, for example, that a file can be transferred to BS2000 and stored there as a PAM file. If you specify *-tff=b*, you must also specify the option *-b* (binary).

s The destination file is to be saved as a sequential file and the record format is to be retained. This allows an ISAM file or PAM file to be fetched from BS2000, for instance.

-tff must not be specified at the same time as *-trf*.

-trf=u Specifies that the file is to be transferred as a sequential file and that the record format of the destination file is to be undefined, i.e. any existing record format of the send file is lost. If the file is being transferred to a BS2000 or z/OS system, one block is written per transfer unit.

-trf must not be specified at the same time as *-tff*.

Neither *-tff* nor *-trf* specified
The destination file is to be stored in the same format as the send file.

-tb=n | **-tb=f** | **-tb=a**
Activates/deactivates tabulator expansion and the conversion of blank lines into lines with one character for a single output send request. The following parameters are provided:

n (on)
Tabulator expansion and blank line conversion are activated.

f (off)
Tabulator expansion and blank line conversion are deactivated.

a (automatic, default value)
Tabulator expansion and blank line conversion are activated if a file is sent to a BS2000, OS/390, or z/OS system.

No tabulator expansion or blank line conversion is performed for outbound receive requests. If *ncopy* is used as a preprocessing command, then tabulator expansion and blank line conversion are always deactivated.

The following parameters *-av, -ac*, *-am,* and *-lq* are provided exclusively for communication with FTAM partners. openFT thus supports the parameters defined in the FTAM standard. These parameters enable you to define the attributes of the destination file while issuing a file transfer request.

These parameters are ignored for requests involving openFT partners, but the file transfer is still carried out.

-av=i | **-av=d**
Indicates the availability of the destination file. This parameter can have one of two values: *immediate* or *deferred*. A file may be *deferred* if it has been archived, for example. The partner is responsible for interpreting the term *deferred*. The FTAM partner conventions must therefore be observed here.

The following values are possible:

i The destination file attribute is set to *immediate*.

d The destination file attribute is set to *deferred*.

av is not available for requests involving FTAM partners that do not support the storage group. In this case, the request is executed, but the entry for *av* is ignored.

-av not specified
The availability file attribute is set to a system-specific default value. In Unix systems, this is the value *immediate*.

-ac=new account
With FTAM partners, this indicates the number of the account to which file storage fees are to be charged. This parameter must be set in accordance with partner system conventions.

ac is not available for requests involving FTAM partners that do not support the storage group. In this case, the request is executed, but the entry for ac *is* ignored.

-am=[r][i][p][x][e][a][c][d] | -am=@rw | -am=@ro

This sets the access rights of the destination file, provided the security group is available.

The following values can be specified:

r, *i*, *p*, *x*, *e*, *a*, *c*, *d*, any combination of these values, *@rw*, or *@ro*.

r means that the file can be read.

r not specified
The file cannot be read.

i means that data units, such as records, can be inserted in the file.

i not specified
No data units can be inserted in the file.

p means that the file can be overwritten.

p not specified
The file cannot be overwritten.

x means that data can be appended to the file.

x not specified
The file cannot be extended.

e means that data units, such as records, can be deleted from the file.

e not specified
No data units can be deleted from the file.

a means that the file attributes can be read.

a not specified
The file attributes cannot be read.

c means that the file attributes can be changed.

c not specified
The file attributes cannot be changed.

d means that the file can be deleted.

d not specified
The file cannot be deleted.

@rw is the short form of the common access rights *read-write* (*rpxeacd*), and thus simplifies input.

@ro is the short form for the common access rights *read-only* (*rac*), and thus simplifies input.

In Unix systems or in BS2000, only the following access rights can be set for a file:

Access mode	Short form	Unix system	BS2000	Access rights
rpxeacd	@rw	rw*	ACCESS=WRITE	read-write
rac	@ro	r-*	ACCESS=READ	read-only
pxeacd		-w*	Only with BASIC-ACL (Access Control List)	write-only
ac		--*	Only with BASIC-ACL (Access Control List)	none

* The x bit is not changed by *ncopy*.

am is not available for requests involving FTAM partners that do not support the security group. In this case, the request is executed, but the entry for *am* is ignored.

-am not specified
The default values of the FTAM partner system apply.

-lq=legal qualification
This specifies a legal qualification for the destination file (similar to a copyright). This may not exceed 80 characters.

lq is not available for requests involving FTAM partners that do not support the security group. The request is executed, but the entry for *lq* is ignored.

-cp=[password]
If a password is required in order to create a file on a remote system, this password must be specified here. It can be up to 64 characters long. A binary password must be specified in hexadecimal format in the form x\'...\' or X\'...\'.
If you do not specify a file creation password, but you do enter a file access password for *-p=password*, the file creation password is identical to the file access password. The file creation password is of no significance when retrieving a file.

password not specified

Specifying *-cp=* causes openFT to query the file creation password on the screen after the command is entered. Your entry is not displayed to prevent unauthorized persons from seeing the password.

Examples

1. The text file *airplane* is sent to the login name *bill* with account number *a1234ft* and password *C'pwd'* in the BS2000 computer with the partner name *bs2r1*, where it is to be printed out.

```
ncopy␣airplane␣bs2r1!%␣bill,a1234ft,C\'pwd\' \
␣-rs="/PRINT-FILE␣airplane,LAYOUT-CONTROL=PARAMETERS\
(,CONTROL-CHARACTERS=EBCDIC)"
```

2. A file is to be fetched from BS2000, where openFT-AC for BS2000/OSD is running, to a Unix system. The file name has been predefined in an FT profile, which can be accessed with the authorization *'onlyforme'*. In the Unix system the file is to be stored under the name *stat.b*. It is to be transferred as an unstructured binary stream. The data is to be compressed for file transfer.

```
ncopy␣-b␣bs2!␣stat.b␣'onlyforme'␣-k
```

3. The text file *letter* is sent to the login name *joe* with the password *pass* in the Unix system with the host name *xserver*. The file should then be printed out in the remote Unix system.

```
ncopy␣letter␣xserver!letter␣joe,,pass␣-rs="lpr␣letter"
```

4. The text file *letter* is sent to the login name *jim* with the password *jimfun* in the FTAM partner with the host name *ftampart*.

```
ncopy␣letter␣ftam://ftampart:102.FTAM.FTAM.FTAM!letter \
␣jim,,jimfun
```

   The FT administrator can use *ftaddptn* to enter the partner in the partner list in order to shorten the command, e.g.

```
ftaddptn ftamp1 ftam://ftampart:102.FTAM.FTAM.FTAM
```

   The *ncopy* command is then:

```
ncopy␣letter␣ftamp1!letter␣jim,,jimfun
```

5. The text file *locfile* is sent to the login name *charles* with the password *secret* in the Unix system *ux1*. There, it is stored under the file name *remfile*. As follow-up processing, the file is printed if transferred successfully; if not, the *prog* program is started in the remote system. This program receives the name of the source file and the message number as parameters. The parameters are specified using variables.

```
ncopy␣locfile␣ux1!remfile␣charles,,secret -r=100 \
␣-rs='lpr remdfile' \
␣-rf='prog %FILENAME %RESULT'
```

   If file transfer is not successful, e.g. because the record length was greater than 100 bytes, follow-up processing is executed as follows:

```
prog remfile 2210
```

6. The *ls* command enables you to view a list of files in a directory on the screen. You want to store this information as a text file in the remote system *wx1* and give this file the name *unix.dir*. The userid is *smith* and the password *any*.

```
ls␣|␣ncopy␣-␣wx1!unix.dir␣smith,,any
```

7. Data is sent from the keyboard to the user *smith* whose computer is *wx1* with the password *any*. The data is stored in the file *MEMO*.

```
ncopy␣-␣wx1!memo␣smith,,any
```

   Then you enter via the keyboard:

```
Will be in headquarters at 4 p.m.
Regards, Johnson
```

   The entry is to be terminated by the <END> or CTRL+D key. The successful transfer is indicated by the message:

```
ncopy: request 65786. File 'STDIN' transferred
```

8. This example shows how to bypass the restriction of follow-up processing commands to 1000 characters in total.

   The text file *finalreport* is sent to the central system *ux1* for storage under the login name *branch1* with password *a-to-z* under the file name *helpfile*. After successful transfer, the file is stored in the directory */home/branch1/file.smith* under the file name *finalreport*, printed out, and appended to the file *file.smith*. The file *file.smith* is then sent to the boss's computer *bosscomp*. In the event of errors, a detailed entry is to be written to the log file *errlog* in the remote system *ux1*.

The restriction is bypassed here by placing the follow-up processing commands in procedures. *succproc* is the procedure for remote follow-up processing if the transfer is successful, and *failproc* is the procedure for remote follow-up processing if the transfer fails.

```
ncopy␣finalreport␣ux1!helpfile␣branch1,,a-to-z\
␣-rs='succproc' \
␣-rf='failproc'
```

If file transfer is successful, the procedure *succproc* is executed in the remote system under the login name *branch1*. This contains the following commands:

```
cp␣helpfile␣/home/branch1/file.smith/finalreport
lpr␣-ws=G005␣-  pb3␣/home/branch1/file.smith/finalreport
cat␣helpfile␣>>␣/home/branch1/file.smith/file.smith

ncopy␣/home/branch1/file.smith/file.smith␣bosscomp!file.smith␣\
secretary,,secret
rm␣helpfile
```

If file transfer is not successful, the procedure *failproc* is executed in the remote system under the login name *branch1*. This contains the following commands:

```
echo "In the event of an error, a detailed message  " >> errlog
echo "should be written to the log file.            " >> errlog
echo "In this case, you can assume that the file    " >> errlog
echo "transfer failed.                              " >> errlog
```

Please note here that the *succproc* and *failproc* procedures must be executable (*rwx------*) in the remote system, or called with *sh* (e.g. *-rs='sh␣succproc'*).

9. Example of the use of preprocessing commands:

   The remote Unix system *ux1* possesses a tar archive *tar.all* under the ID *karlotto* with the password *secret*. The file *file.1* is to be retrieved from this tar archive into the local system and saved in the local file *file.loc*.

   ```
   ncopy␣ux1!"|ft_tar␣-xOf␣tar.all␣file.1"␣file.loc \
   ␣karlotto,,secret
   ```

   ft_tar -xOf retrieves the file from the archive and writes it to *stdout*. The file *file.1* is then therefore not available under the remote ID.

10. Example of the use of postprocessing commands:

    The local file *file* is to be entered in the tar archive *tar.all* under the name *file.x*. The tar archive *tar.all* is located on the remote computer *win1* under the transfer admission *tarremote*. After being entered in the tar archive, the file is to be deleted in the remote system.

```
ncopy␣file␣win1!"|cp␣%TEMPFILE␣file.x;ft_tar␣-uf␣tar.all \
␣file.x␣--remove_files"␣tarremote
```

11. Example for illustrating the use of preprocessing and postprocessing commands.

    The local directory *dir* and all its files are to be transferred to the remote Unix host with the symbolic name *ftunix*. The current openFT version is also running on the remote host. After the transfer, *dir* should be available on the remote system under the user ID that owns the *copydire* transfer admission.

```
ncopy␣"|ft_tar␣-cf␣-␣dir"␣ftunix!"|ft_tar␣-xf␣-␣"␣copydire␣-b
```

    The *dir* directory must be located on the local computer in $HOME. Please note that no file name prefixes may be defined in the profile.

12. Example of the use of preprocessing and postprocessing commands:

    At the remote computer *ux1*, you first want to compress the remote file *remfile* under the user ID *karlotto* with the password *secret* (using the command *compress -c remfile*). The result is transferred and written to the local system's standard output (-). Here, the output is transferred via a pipe to the *uncompress -c* command and saved in the local file *locfile*.

```
ncopy␣-b␣ux1!"|compress␣-c␣remdate"␣"\
|uncompress␣-c>locfile"␣karlotto,,secret
```

    If the command is rejected with `Remote System: Exitcode 2` in the case of preprocessing/postprocessing then the cause may lie in the remote system's *compress* command. In some Unix systems, the command supplies return code 2 even though it was successful.

    You can avoid this problem by extending the preprocessing command with 'exit 0':

```
ncopy␣-b␣ux1!"|compress␣-c␣remdate;exit 0"␣"\
|uncompress␣-c>locfile␣karlotto,,secret
```

13. Example for FTP connection

    In the remote system with the host name *wini2* there is only one FTP server. The file *all_files* under the ID *user1* with the password *usrpass* is to be fetched into the local system. Here, it is to be stored in the directory *save_files* under the partner-specific name *wini2.all_files*.

```
ncopy␣ftp://wini2!all_files␣save_files/%PARTNER.all_files \
␣user1,,usrpass
```

6 openFT-Script Commands

The openFT-Script commands are used to start and administer openFT-Script requests. The requests themselves are stored in a text file in the form of XML statements. These XML statements are described in the "openFT-Script Interface" manual.

6.1 Overview of the openFT-Script commands

Starting and ending openFT-Script requests

ftscript Starts an openFT-Script request

ftcans Cancels an openFT-Script request

ftdels Deletes an openFT-Script request

Displaying openFT-Script requests and openFT-Script activities

ftshws Displays openFT-Script requests

ftshwact Displays the activities of an openFT-Script request

FT administrators can also use the *ftsetjava* command to administer the link to the Java executable, see "openFT System Administrator Manual" and the online help system.

As FT administrator, you can view, cancel and delete all the openFT-Script requests in the system and monitor the activities associated with all the openFT-Script requests. Users without administrator rights can only administer their own openFT-Script requests.

6.2 ftcans - Cancelling an openFT-Script request

ftcans allows you to cancel openFT-Script requests that have not yet been concluded. You can cancel either a specific openFT-Script request or all the openFT-Script requests for a user. This also cancels any file transfer requests started by the specified openFT-Script requests which are currently running. This may take a little time. The status of the openFT-Script request is then set to "cancelled" to prevent any restart.

If the openFT-Script request that is to be cancelled is currently being processed then the following message is output at *stderr*:

`ftcans: Cancellation request for ftscript id` *`ftscript id`* `started`

If the request has been started but not yet processed then the following message is sent to *stderr*:

`ftcans: ftscript id` *`ftscript id`* `cancelled.`

Format

ftcans -h |
[-u=<user ID 1..32>]
<ftscriptid> | @a

Description

-h Outputs the command syntax on screen. Any specifications after *-h* are ignored.

-u=user ID
User ID under which the search for the openFT-Script request that is to be cancelled is performed.

Only the FT administrator may input a user ID.

The default value is the calling party's user ID.

ftscriptid
Identification of the openFT-Script request. This is output if the openFT-Script request is started via an ftscript command.

@a means that all the user's openFT-Script requests are to be cancelled.

Return code

0	OK
4	Syntax error
51	Error while outputting an Ftscript user
54	Ftscript ID not found
250	Internal error

6.3 ftdels - Deleting an openFT-Script request

The specified, completed openFT-Script request is deleted from the user's directory or all completed openFT-Script requests are deleted from the user's directory.

No more information is subsequently available for deleted requests. A *ftshws* or *ftshwact* command with this *ftscriptid* is rejected since it no longer exists.

Before an openFT-Script request can be deleted, it must have been completed, i.e. *ftshws* must indicate the status T, F or C.

Since *ftcans* is not a synchronous command, it may be necessary to wait for the status C (cancelled) to arise before a subsequent *ftdels*.

If no *ftdels* is issued for an openFT-Script request then this is automatically deleted when the retention period (3 days) expires.

Format

ftdels -h |
[-u=<user ID 1..32>]
<ftscriptid> | @a

Description

-h Outputs the command syntax on screen. Any specifications after *-h* are ignored.

-u=user ID
User ID under which the search for the openFT-Script request that is to be deleted is performed.

Only the FT administrator may input a user ID.

The default value is the calling party's user ID.

ftscriptid
Identification of the openFT-Script request. This is output when the openFT-Script request is started via an *ftscript* command.

@a means that all the user's openFT-Script completed requests are to be deleted.

Return code

0	OK
4	Syntax error
51	Error while outputting an Ftscript user
54	Ftscript ID not found
56	openFT-Script has not completed
250	Internal error

6.4 ftscript - Starting an openFT-Script request

The *ftscript* command checks the specified script file and executes the statements it includes. The script file must contain a valid XML document which corresponds to the schema for the openFT-Script interface. It must also be possible to read the file using the caller's ID. The maximum number of users who may be owner of openFT-Script requests is 1024. This includes requests that are terminated but not yet deleted.

If errors occur during verification then the script file is not started and the errors are output at *stderr*.

If the script file starts correctly then the following message is output at *stderr*:

`ftscript: started successfully. Id:` *ftscript id*

Information about the openFT-Script request is stored in the internal openFT user memory during execution and through to expiry of the retention period. As a consequence, users can view the output *ftscript-id* in order to obtain information about the status and success of the operation.

ftscript is restartable, i.e. the processing of the openFT-Script request is ensured even after a system failure.

Format

ftscript -h |
[-t]
<Ftscript file name>

Description

-h Outputs the command syntax on screen. Any specifications after *-h* are ignored.

-t Diagnostic information (a trace) is created.

Ftscript file name
Name of the script file which contains the XML statements for the openFT-Script request that is to be run.

Return code

0	OK
4	Syntax error
50	Ftscript process could not be started
52	Maximum number of Ftscript users (1024) exceeded
55	Ftscript ID not found
250	Internal error

6.5 ftshwact - Displaying the activities associated with an openFT-Script request

Outputs information about the activities of the specified openFT-Script request.

Format

ftshwact -h |
[-csv]
[-a=<ID of the activity> | -d=<Level depth 1...> | -c=<Chapter>]
[-st=[W][R][T][F][K][D][C]]
[-u=<user ID 1..32>]
<ftscriptid>

Description

-h Outputs the command syntax on screen. Any specifications after *-h* are ignored.

-csv The information is output in CSV format.

-a=ID of the activity
Only the specified activity is displayed.

You may also indicate a specific instruction in a request.

An activity's ID can be determined using a preceding *ftshwact* command (without the *-a* option). This means that you can view the status of the activity later.

-d=Level depth
Depth of the levels to be displayed.

All activities whose *activity IDs* have a level not greater than the specified level number are displayed. The level number is the number of index numbers separated by dots.

Examples:
From a request with activity IDs 1, 1.2, 1.2(1).1, 1.2(1).2, 1.2(2).1, 1.2(2).2 and1.3 the option *-d*=2 selects the activities with the activity IDs 1, 1.2 and 1.3.

-c=Chapter

Chapter corresponding to the activities to be displayed.

Those activities are output that are one level below the activity with the activity ID specified as the chapter.

In the above example, these are for *-c=1*: 1.2 and 1.3; for *-c=1.2*: 1.2(1).1, 1.2(1).2, 1.2(2).1 and 1.2(2).2.

-st=[**W**][**R**][**T**][**F**][**K**][**D**][**C**]

Display activities with the specified status. You can specify multiple statuses one after the other, e.g. *-st=WRT*.

Activity 1 is always output since it displays the execution status of the entire script.

-u=user ID

User ID under which the specified request is searched for.

Only the FT administrator may input a user ID.

The default value is the calling party's user ID.

ftscriptid

Identification of the openFT-Script request. This is output when the openFT-Script request is started via an *ftscript* command.

Return code

0	OK
4	Syntax error
51	Error while outputting an Ftscript user
53	Ftscript section not found
54	Ftscript ID not found
250	Internal error

Description of the output

Output is possible in tabular form and in CSV format.

It should be noted that for activities which have not yet been started, the output from the *ftshwact* command is usually incomplete since the references present in the request have not yet been resolved and it is not therefore possible to enter all the desired output values. In particular, file and directory names in reference specifications are not fixed until runtime since they may be dependent on the operating system.

Output in table format

The processing level of the activities is displayed in four columns:

Id Unique identification of the activity within the request. This can be converted into an Xpath which mirrors the position of the activity in the tree which is statically predefined by the XML script.

Dynamic information is simply added for the *foreach* nodes (sequence number in the *foreach* loop).

For more detailed information, see the description of the XML statements for the openFT-Script interface.

Sta Status of the statement. The following status identifiers are possible:

W (waiting)	The activity has not yet been started.
R (running)	The activity has been started but has not yet been terminated.
T (terminated)	The activity has been terminated without errors.
F (failure)	The activity has been terminated with an error.
K (killed)	The activity was cancelled by means of a faulthandler or an *ftcans* command.
D (dead)	The activity no longer starts due to a previous error.

In the case of the *ftscript* activity (first activity in an openFT-Script request), a distinction is made between the following statuses:

I (interrupted)	The request was interrupted, e.g. due to a system crash.
C (cancelled)	The request was cancelled with *ftcans*.

X (cancelling) The request is currently being cancelled due to an *ftcans* command.

F (failure) Is only displayed for the *ftscript* activity if the error was not handled by a *faulthandler*.

In the case of activities with the status F and *faulthandler* activities, the cause of the error is output in clear text in an additional line.

Activity

Activity name. The names are based on the openFT-Script language but may be truncated in some cases, e.g. *faulthdlr* instead of *faulthandler*.

foreach is designated in accordance with the value of the execute attribute as *foreach_P* (parallel) or *foreach_S* (sequential).

TransferFile is designated as *sendFile* or *rcvFile* (=receive File) depending on the direction of transfer.

ActivityObject

The content of this column depends on the activity in question, see the table below.

Activity	**ActivityObject**	**Meaning**
ftscript	<scriptPath>	Complete path name of the original file with the XML statements.
empty	-	
foreach_P	<contextObject>	Context object which assumes the value of the current list element
foreach_S	as foreach_P	
parallel	-	
sequence	-	
sendFile	Specifies the remote file in the following form:	
	<partner>!<file name>	Partner with file name if both are known.
	*unknown!<file name>	if the partner is not yet known.
	*unknown!*unknown	if both are not yet known.

Activity	ActivityObject	Meaning
sendFile (*cont.*)	<partner>!*ref(<contextId>)	if *contextId* = *foreach contextObject* and the resolution is not yet known because it has not yet been passed through.
	<file name>	in the case of requests which have already been started, this is the name specified in the FT request. In the case of requests which have not yet been started, this name is derived from the operating system-specific name specified in the XML file (e.g. unixName) and extended by the directory specifications.
rcvFile	as sendFile.	
deleteFile	specifies the remote file as in sendFile (with partner), if the file is local without partner:	
	<file name>	like *sendFile*, is determined from the FT request in the case of requests that have already been started, and from the XML file in the case of requests that have not yet started. A local file name would be output as an absolute file name in the case of a started request and as a relative path name in the case of an as yet unstarted request.
	*unknown!<file name>	if it is not known if the file is local when a file object is referenced.
createDir	<partner>! <directory-name>	Partner with directory name if both are known.
	*unknown! <directory-name>	if the partner is not yet known.
	*unknown!*unknown	if both are not yet known.

Activity	**ActivityObject**	**Meaning**
createDir (*cont.*)	<partner>! *ref(<contextID>)	if *contexId* = *foreach contextObject* and the resolution is not yet known because it has not yet been passed through.
	<directory-name>	if the directory is local. In this case, as for *sendFile*, the name for already started requests is determined from the FT request and for requests which have not yet been started, from the specifications in the XML file. A local file name would be output as an absolute file name in the case of a started request and as a relative path name in the case of an as yet unstarted request.
deleteDir	as createDir.	
listDir	as createDir.	
execScript	32 characters.	Contains the first 32 characters of the command that is to be executed. For security reasons, the user should make sure that the first 32 characters do not contain any confidential parameters.
fault	<faultcode>	Error code specified by the user.
faulthdl	<triggering activity id>: <special faultcode>; <general faultcode>	

Output in CSV format

```
Id;State;Activity;ActivityObject;Partner;AddInfo;NrElements;
StartTime;Error
```

The output contains the following information:

Id	See table format on page 364.
State	See table format on page 364.
Activity	See table format on page 365.
ActivityObject	See table format, enclosed in double quotes, otherwise: - the path name is output without partner specifications - only the *faultcodes* are output for the *faulthandler* activity.
Partner	In the case of path-related activities, the partner or partner specification that would be present in front of the path name in table format, enclosed in double quotes. Otherwise empty.
AddInfo	For *sendFile* and *rcvFile*: TID, enclosed in double quotes if the activity has already started. Otherwise empty. For *faulthdl*, the triggering *activity-Id* enclosed in double quotes. Otherwise empty.
NrElements	In the case of a started *foreach*: number of loop passes. In the case of a started *parallel* or *sequence*: number of sub-activities.
StartTime	Start time in the format yyyy-mm-dd hh:mm:ss
Error	In the case of requests with the status F, case of error in clear text enclosed by double quotes. Otherwise empty.

6.6 ftshws - Display openFT-Script requests

Outputs information about the status of a user's openFT-Script requests. You can also specify a *ftscriptid* in order to select a specific openFT-Script request.

Format

ftshws -h |
[-csv]
[-t]
[-v]
[-st=[W][R][T][F][I][C][X]]
[-u=<user ID 1..32> | @a]
[<ftscriptid>]

Description

-h Outputs the command syntax on screen. Any specifications after *-h* are ignored.

-csv The information is output in CSV format.

-t The openFT-Script requests are displayed sorted on generation time, beginning with the last request.

By default, the requests are displayed in alphabetical order.

-v Diagnostic information is also output (verbose).

If *-v* is specified then, in the case of openFT-Script requests which terminate with an error, the cause of the error is output in a second line after the tabular information.

In CSV format, the *-v* option is ignored.

-st=[**W**][**R**][**T**][**F**][**I**][**C**][**X**]
displays openFT-Script requests with the specified status, see State field in "Output in table format" on page 370.

You can specify multiple statuses one after the other, e.g. *-st=WRT*.

-u=user ID | **@a**
User ID for which openFT-Script requests are output or under which the specified request is searched for.
Only the FT administrator may specify a user ID or *@a* (all user IDs).

The default value is the calling party's user ID.

ftscriptid
: Identification of the openFT-Script request. This is output if the openFT-Script request is started via an *ftscript* command.

  By default, all the user's openFT-Script requests are displayed.

Return code

0 OK

4 Syntax error

51 Error while outputting an Ftscript user

54 Ftscript ID not found

250 Internal error

Output in table format

The processing level of the openFT-Script requests is displayed in four columns:

User User ID under which the request was started.

Ftscriptid
: Unique identification of the request. The identification is returned by the *ftscript* command.

Sta Indicates the processing status, where:

W (waiting)	The request has not yet been started.
R (running)	The request has been started but has not yet been terminated.
T (terminated)	The request has been terminated without errors.
F (failure)	The request has been terminated with errors.
I (interrupted)	The request was interrupted, e.g. due to a system crash.
C (cancelled)	The request was cancelled with an *ftcans* command.
X (cancelling)	The request is currently being cancelled due to an *ftcans* command.

FtscriptFileName
: Path name of the script file.

If the status F and the *-v* option are specified then the cause of the error is output in clear text in another column.

Output in CSV format

```
User;Ftscriptid;State;CreationTime;FtscriptFileName;Error
```

The output contains the following information:

User	User ID under which the request was started.
Ftscriptid	Unique identification of the request. The identification is returned by the *ftscript* command.
State	See table format (Sta).
CreationTime	Time at which the openFT-Script request was created, in the format yyyy-mm-dd hh:mm:ss.
FtscriptFileName	Path name of the script file.
Error	Cause of error in clear text in the case of openFT-Script requests with status F, otherwise empty.

User, *Ftscriptid*, *FtscriptFileName* and, if applicable, *Error* are output enclosed in double quotes.

7 Program interface

openFT offers the following program interfaces on Unix systems:

- C program interface
- JAVA program interface

7.1 Programming with C

You can use the C program interface to incorporate the functionality of openFT in your own C programs:

– synchronous file transmission

– asynchronous file transfer

– managing and deleting asynchronous file transfer requests

– determining file attributes in the remote system

– deleting files or directories in the remote system

– creating directories in the remote system

These functions which are available to the openFT user can be used in programs to automate sequences. The program interface also provides monitoring and error handling mechanisms.
In addition, the program interface has a function call which you can use determine the properties of the program interface. You can use this call to check the properties and thus render your programs insensitive to changes in later versions.

The following overview is useful for quick orientation with respect to which C program calls are available for which tasks. The corresponding FT commands which the user can work with on the shell level are indicated in brackets (see the chapter “openFT commands for the user” on page 139).

You can find a description of the C functions in the manual “openFT for Unix and Windows systems - C Program Interface”.

7.2 Programming with JAVA

Use this program interface to include the following openFT functions in the JAVA programs you create:

- synchronous file transfer
- asynchronous file transfer
- administer and delete asynchronous file transfer requests
- transmit file attributes in remote systems
- create directories in remote systems
- delete files or file directories in remote systems

These functions are available to openFT users and can be used in JAVA programs to automate procedures.

You will find the JAVA DOCS in the directory */opt/openFT/java/doc*.

To translate your program, you must also specify the class path */opt/openFT/java/openFTapi.jar*.

The class path */opt/openFT/java/openFTapi.jar* will also be needed for the program to run, in addition to the java.library.path */opt/openFT/java*.

After installing openFT, you will find sample programs (*Sample[1..3].java*) in the directory */opt/openFT/samples.*

Example

Translating the sample and running the program *Sample1.java*:

1. Copying *Sample1.java* into the current directory

```
cp /opt/openFT/samples/Sample1.java .
```

2. Translating *Sample1.java*

```
javac –classpath /opt/openFT/java/openFTapi.jar:.
       Sample1.java
```

3. Running *Sample1.class*

```
java –cp /opt/openFT/java/openFTapi.jar:. \
     –Djava.library.path=/opt/openFT/java Sample1 dat1 dat2
```

8 What if ...

8.1 Actions in the event of an error

If, in spite of precautions, an error occurs which neither the FTAC administrator nor the FT administrator can rectify, please contact your local contact partner. In order to simplify error diagnosis, you should provide the following documents:

– an exact description of the error situation and information as to whether the error is reproducible;

– specification of the platforms on which the involved file transfer products run in the local and in the partner system (e.g. Linux, Solaris, Windows XP, BS2000/OSD, ...)

– the version number of the file transfer product in the local and in the partner system and if applicable, the version number of FTAC installed there;

– diagnostic information (which is created with the FT command *ftshwd*);

– if available, the FTAC, FT and ADM log records (which are output with the FT command *ftshwl*);

– if available, the openFT trace file;

– for errors related to a specific FT profile, a printout of the profile *(ftshwp␣profilename␣-l)* and a printout of the admission sets *(ftshwa␣@a)*.

8.2 Locked transfer admissions - possible causes and remedies

If FTAC rejects a file transfer request on account of an invalid transfer admission, the cause may be one of several:

– No transfer admission was defined when the FT profile was created or modified.

– A user wished to create an FT profile with a transfer admission which was already assigned to a different FT profile on the computer. If the relevant FT profile is marked as private, the transfer admission becomes invalid. At the same time, the values for date, scope (public/private) and validity (*-d*, *-u* and *-v*) are set to the default values.

– The FTAC administrator modifies an FT profile for a user without knowledge of the complete login admission. In this case, the transfer admission remains valid, but is locked.

– The FT profile was imported by an FTAC administrator who is not the FT administrator. It is therefore locked automatically.

– The FT profile was locked explicitly.

– The period during which the transfer admission may be used has expired.

The detailed output of the *ftshwp* command displays the cause of an invalid transfer admission using the additional parameter *TRANS-ADM*. The possible values for this parameter, the meanings and counteractions are shown in the table “TRANS-ADM=” on page 308.

9 Messages

The openFT messages are sent to you as a result code (shell variable $?) and as text to the screen *stderr*.

The messages appear in the language that is set for openFT (English or German). Please refer to section “Switching language interfaces” on page 64 for a detailed description how you can switch the language.

If multiple file transfers are running in parallel, you can use the request ID to assign the error message to the correct file transfer.

<local file> or <remote file> specifies the file name.

<Request id> specifies the number of the file transfer request. openFT informs you of this number on confirmation of request receipt.

The message texts and result codes have changed in openFT V10.0 compared to V8.1. To permit the compatible execution of existing shell procedures which evaluate result codes, openFT makes it possible to return compatible exit codes by setting the OPENFTOLDXC environment variable. If conversion to a compatible exit code is not possible, exit code 248 is issued.

This option of setting the environment variable OPENFTOLDXC will be supported for the last time in V11.

There follows a description of the error messages output by openFT together with the associated exit codes, meanings and measures where appropriate.

The description has the following format:

exit code Message text
meanings and measures as appropriate

9.1 openFT messages

9.1.1 Messages applying to all commands

0 The command was successful

3 The command was cancelled as the result of a response to a query

4 A syntax error occurred during command processing

225 Information output canceled

Meaning:
A show command was interrupted, for example.

Measure:
Repeat the command.

226 Monitor file contents inconsistent

Meaning:
The command cannot be accepted because the contents of the specified monitor file are inconsistent.
Possible reason: The monitor file was accessed by the user in a mode other than read mode while it was monitoring an FT request.
The contents of the monitor file can no longer be used.

227 Monitor file not in use by openFT

Measure:
Correct the name of the job variable and repeat the command.

228 Monitor file not found

Measure:
Correct the name of the job variable and repeat the command.

236 Current instance '<instance>' no longer found

Meaning:
The command was rejected. The instance '<instance>' could not be found.

250 An internal error occurred during command processing

251 Command aborted with core dump

253 Current openFT instance is invalid

Meaning:
During command processing a defined instance was found to be invalid

255 ftexec/ftadm command failed

Meaning:
Remote execution of the command with ftexec failed

9.1.2 Messages applying to file transfer, file management and remote administration commands

All the messages listed below, with the exception of the message with exit code 5, can also be output during logging. In this case, however, the specified code is increased by 2000, e.g. 2169 instead of 169.

5 Request <Request id>. File '<local file>' transferred

Meaning:
The file transfer request <Request id> has been successfully completed. Follow-up processing has been started for both the local system and the remote system, as requested, provided no error occurred. Local errors are indicated as a message.

14 No file attribute changes requested

Meaning:
No further file attributes besides the file name were specified.

Measure:
Enter the desired file attributes in addition to the file name.

15 openFT is not authorized to execute requests for this user

16 Directory '<local file>' is not empty

17 File attributes do not match request parameters

Meaning:
The specified attribute combination is not permissible.

Measure:
Specify a permissible combination.

18 Attributes could not be modified

Meaning:
The properties of the file could not be changed as specified in the command. The following reasons are possible:

For the remote file:
– No access rights to the file.
– The required combination of access rights is not supported by the remote system.
– If the remote system is a BS2000: the file is protected by ACL.

For the local file:
– No access rights to the file.
– The requested transfer attributes are not compatible with the BS2000 properties of the file.

19 '<local file>' could not be created

Meaning:
The command was not executed because the file owner and user requesting the creation of a receive file are not the same.

Measure:
Match the user ID in the receiving system's transfer admission to the ID of the receive file's owner.
Repeat the command.

20 '<local file>' not found

Meaning:
The command was not executed because the send file is not in the catalog or on a volume of the local system. The command was not executed because either the send file is not/no longer or the receive file is no longer in the catalog or on a volume of the relevant system.

Measure:
Correct the file name, read in file from tape or restore the send file.
Repeat the command.

21 CCS name unknown

Meaning:
The request could not be carried out because the CCS names of the send and receive files could not be mapped to each other or because the partner system does not support the transparent receipt of files.

22 Higher-level directory not found

Meaning:
In the case of a receive request, the local file could not be created because the specified path does not exist.

Measure:
Create or correct the path for the receive file and repeat the command.

23 '<local file>' already exists

Meaning:
The command was not executed because an existing receive file cannot be created again with option *-n*.
Option *-n* may also have been set due to a restriction in the access authorization used.

Measure:
Either delete the receive file and repeat the command, or repeat the command specifying option *-o* or using different access authorization.

24 Transfer of file generation groups not supported

Meaning:
The command was not executed because the FT system only transfers single file generations.

Measure:
Repeat the command using the name of a single file generation.

25 Error accessing '<local file>' <2>

Meaning:
<2>: DMS error, possibly the transfer ID.
The FT system continues to run after the message has been issued.

Measure:
Take the appropriate action in accordance with the error code.

26 Resulting file name'<local file>' too long

Meaning:
The relative file name was specified in the transfer request.
The absolute file name completed by openFT is longer than permitted.

Measure:
Shorten the file name or path and repeat the command.

27 No file or directory name specified

Meaning:
The command was not executed because the file name was neither specified explicitly nor by the 'transfer admission' used.

Measure:
Repeat the command, specifying the file ID explicitly or a transfer admission that defines the file ID.

28 Invalid management password

29 '<local file>' not available

Meaning:
The command was not executed because the volume for either the send file or the receive file is not mounted, unknown or reserved, or the file extends over more than one private disk.

Measure:
Inform the operator if necessary. Repeat the command.

30 Home directory not found

31 Renaming not possible

32 Not enough space for '<local file>'

Meaning:
The command was not (fully) executed because
– the permissible storage space on the receive system is used up for the user ID specified in transfer admission, or
– the send file contains too long a sequence of empty blocks, or
– the primary and/or secondary allocation of the password-protected receive file is too small.

The receive file can not be created/extended after the problem occurs.

Measure:
Take the appropriate action depending on the cause of the error:
– delete all files no longer required on the receive system, or
– ask the system administrator to allocate more storage space, or
– remove empty blocks from the send file, or
– increase the receive file's primary/secondary allocation.

If option *-e* is specified, restore the receive file.
Repeat the command.

33 File owner unknown

Meaning:
The command was not executed because the owner of either the send file or the receive file was not defined in the local system or because the file owner and the user requesting the creation of a receive file are not the same.

Measure:
Define the file owner, correct transfer admission or file name.
Repeat the command.

34 Invalid file password

Meaning:
The command was not executed because the password for the send file or the receive file is missing or incorrect.

Measure:
Correct the password in the file description or the command.
Repeat the command.

35 File locked to prevent multiple access

Meaning:
The command was not executed because either the send file or the receive file is already locked by another process to prevent it from being updated simultaneously.

Measure:
Repeat the command later or unlock the file.
After a system crash you may need to verify files that are not closed correctly.
If the lock is caused by an FT request, it will be canceled automatically when the request is finished.

36 Retention period of file not yet expired

Meaning:
The command was not executed because the retention period protecting the receive file against overwriting has not yet expired (RETENTION PERIOD).

Measure:
Correct the transfer direction, retention period or file name.
Repeat the command.

37 '<local file>' is read only

38 File structure not supported

39 Syntax error in resulting file name '<local file>'

Meaning:
The local file cannot be accessed because, for example, the absolute file name is too long.

Measure:
Shorten the path or file name. Repeat the command.

40 Transparent file transfer not supported

Meaning:
The request could not be carried out because the CCS names of the send and receive files cannot be mapped to each other or because the partner system does not support the receipt of files in a transparent format.

41 Request queue full

Meaning:
The command was not executed because the maximum number of permissible file transfer requests has been reached.

Measure:
Notify the FT administrator. Repeat the command later.

42 Extension of file not possible for transparent transfer

Meaning:
The command could not be executed because it is not possible to add to a file in a transparent transfer.

Measure:
Start transfer without option *-e*.

43 Access to '<local file>' denied

Meaning:
The command was not executed because either the send file or the receive file only permits certain access modes (e.g. read only).

Measure:
Correct the file name or file protection attributes. Repeat the command.

44 Follow-up processing exceeds length limit

Meaning:
Prefix + suffix (from prof) + local follow-up processing together are too long.

Measure:
Shorten the follow-up processing, or use procedures.
Repeat the command.

45 Processing admission invalid

Meaning:
The command was not executed because the specifications in one of the PROCESSING-ADMISSION operands were incorrect.

Measure:
Define the required PROCESSING ADMISSION or correct it.
Repeat the command if necessary.

46 Local transfer admission invalid

Meaning:
The command was not executed because the specifications in one of the transfer admission operands were incorrect.

Measure:
Define the required transfer admission or correct it.
Repeat the command if necessary.

47 Request rejected by local FTAC

Meaning:
The command was not executed because the request was rejected by the FTAC due to a lack of authorization.

Measure:
Use the return code in the log record to determine and remove the cause.
Repeat the command.

48 Function not supported for protocol '<partner protocol type>'

Meaning:
The desired function is not available for the selected protocol.

Measure:
Select a different protocol.

49 Remote follow-up processing not supported

Meaning:
Remote follow-up processing is only available for the openFT protocol.

Measure:
Select a different protocol, or specify follow-up processing by means of an FTAC profile.

50 Data integrity check not supported

Meaning:
The partner system does not support the data integrity check function.

Measure:
Repeat the request without a file integrity check.

51 User data encryption not possible for this request

Meaning:
The partner system does not support the data encryption function.

Measure:
Repeat the request without data encryption or install openFT-CR (or have it installed) on the remote system.

52 Administration request rejected by remote administration server

Meaning:
The administration request was rejected by the remote administration server because it clashes with the settings in the configuration file of the remote administration server.

The ADM administrator can determine the precise reason for rejection from the associated ADM log record on the remote administration server.

Possible reason codes:

7001 The administrator ID is invalid. It was not possible to determine a valid administrator ID from the user ID or the profile name in the configuration data of the remote administration server.

7002 The routing information is invalid. The specified openFT instance specified in the routing information could not be found in the configuration data of the remote administration server.

7003 The specified remote administration command is invalid. The remote administration server rejects the specified command because it is not a supported remote administration command.

7101 Infringement against the access rights list. On checking the access rights, the system identified that the administrator ID has not been assigned the necessary rights in the configuration data of the remote administration server to be able to execute the valid remote administration command on the specified openFT instance.

7201 Infringement against the maximum command length. In particular in the case of BS2000 commands, the remote administration server replaces the shortest command names, which are guaranteed by openFT, by the full command names. If this replacement of the command name causes the entire remote administration command to become longer than the maximum command length of 8192 characters, the command is rejected.

Measure:
Have the ADM administrator carry out the necessary adjustments to the configuration data or check the command. Repeat the changed command if necessary.

54 Invalid command

Meaning:
The specified command is not a command that is permitted to be executed on the specified system using the remote administration facility.

Measure:
Specify an admissible command or add the missing routing information. Repeat the command.

55 Transfer of protection attributes not supported

56 Syntax error in partner name '<partner>'

57 openFT is not authorized to execute administration requests

Meaning:
openFT is not (no longer) authorized to process administration requests. This is, for example, the case if a remote administration server has been demoted to a normal server (*ftmodo -admcs=n*) or if commands that are only allowed to be executed on a remote administration server are processed by an openFT instance that has not been configured as a remote administration server.

70 Request <Request id>. openFT is no longer authorized to execute requests for this user

71 Request <Request id>. User data encryption not installed

Meaning:
The user data encryption function cannot be used unless openFT-CR is installed.

Measure:
Use openFT-CR.

72 Request <Request id> has been canceled

Meaning:
The FT request was canceled because the *ftcanr* command was specified, or the time specified in the transfer request has been reached.

Follow-up processing has been started for the local system, provided no error occurred. Follow-up processing is started for the remote system once all the resources are allocated. Local errors are indicated by the message FTR0050 at the start of follow-up processing.

73 Request <Request id>. Encryption error

Meaning:
Encryption not possible.

74 Request <Request id>. '<local file>' could not be created

Meaning:
The command was not executed because the file owner and user requesting the creation of a receive file are not the same.

Measure:
Match the user ID in the receive system's transfer admission to the ID of the receive file owner.
Repeat the command.

75 Request <Request id>. Higher-level directory no longer found

76 Request <Request id>. I/O error for '<local file>'

Meaning:
The file can no longer be accessed. It may have been deleted during a transfer.

Measure:
Repeat the request.

77 Request <Request id>. File now locked to prevent multiple access

Meaning:
The command was not executed because the send file or the receive file is already locked by another process so that it cannot be simultaneously updated.

Measure:
Repeat the command later or unlock the file.
After a system crash you may need to verify files that are not closed correctly.
If the lock is caused by an FT request, it will be released automatically when the request is finished.

78 Request <Request id>. '<local file>' no longer available

Meaning:
The command was not executed because the volume for either the send file or the receive file is not mounted, unknown or reserved, or the file extends over more than one private disk.

Measure:
Inform the operator if necessary. Repeat the command.

79 Request <Request id>. '<local file>' no longer found

Meaning:
The local send or receive file can no longer be accessed because, for example, it was deleted during an interruption of the openFT system.

Measure:
Restore the file.
Repeat the command.

80 Request <Request id>. Home directory no longer found

81 Request <Request id>. '<local file>' gets no more space

Meaning:
The command was not (any further) executed because
- the permissible storage space on the receive system for the user ID specified in transfer admission has been used up, or
- the send file contains too long a sequence of empty blocks, or
- the primary and/or secondary allocation of the password-protected receive file is too small.

The receive file can not be created/extended once this problem occurs.

Measure:
Take the appropriate action depending on the cause of the error:
– delete all files no longer required on the receive system, or
– ask the system administrator to allocate more storage space, or
– remove empty blocks from the send file, or
– increase the receive file's primary/secondary allocation.

If option *-e* is specified, restore the receive file.
Repeat the command.

82 Request <Request id>. File owner no longer known

Meaning:
The command was not executed because the owner of the send file or receive file is not defined on the relevant system or because the file owner and the user who wants to create a receive file are not the same.

Measure:
Define the file owner, or correct transfer admission or file name.
Repeat the command.

83 Request <Request id>. Pre-/post-processing error

Meaning:
The command executed as part of local pre-/post-processing returned an exit code other than 0.

Measure:
Correct and repeat the command.

84 Request <Request id>. Exit code <2> for pre-/post-processing

Meaning:
The command executed as part of local pre-/post-processing returned the exit code <2>.

Measure:
Correct the command using the exit code <2> and issue it again.

85 Request <Request id>. File password no longer valid

Meaning:
The command was not executed because the password for send file or the receive file is missing or incorrect.

Measure:
Correct the password in the file description or the command.
Repeat the command.

86 Request <Request id>. '<local file>' is now read only

87 Request <Request id>. File structure error

Meaning:
The command was not executed due to a file structure error.
File structure errors include:
- The attributes of the send file are incomplete.
- The data of the send file is incompatible with its structure attributes.
- The records of the send file are too long.
- If *-e* is specified, the send file and receive file have different structures (e.g. fixed-/variable-length records).
- The send file or receive file is a member of an old LMS library (not PLAM).
- The source file has an odd block factor (e.g. BLKSIZE=(STD,1)) and the receive file is to be stored on an NK4 pubset.

Measure:
Correct the file or file attributes. If option *-e* is specified, restore the receive file. Repeat the command.

88 Request <Request id>. NDMS error <2>

Meaning:
The request was rejected because the partner system currently does not have the resources available to accept requests.

Measure:
Repeat the request a little later.

89 Request <Request id>. Recovery failed

Meaning:
The restart attempts were unsuccessful (for example, a pre-/post-processing command could not be completed before the termination of openFT).

Measure:
Repeat the command.

90 Request <Request id>. Error in file transfer completion

Meaning:
An error occurred during the final phase of the file transfer.
If it was a long transfer, the recipient is advised to check if the file has still been transferred correctly. However, error follow-up processing will be started if it was specified.

Measure:
Repeat the request, if necessary.

91 Requests only partially completed; <1> of <2> files were transferred

Meaning:
In the case of a synchronous send request with wildcards, not all files were successfully transferred.

Measure:
Transfer unsuccessfully transferred files again.

92 Request <Request id>. Access to '<local file>' no longer permissible

93 Request <Request id>. FTAM error <2>

94 Request <Request id>. Retention period of file not yet expired

95 Request <Request id>. Extension of file not possible for transparent transfer

96 Request <Request id>. File structure not supported

97 Request <Request id>. Resulting file name '<local file>' too long

99 Request <Request id>. Transfer of protection attributes not supported

108 Request <Request id>. Remote system not accessible

Meaning:
The command could not be accepted because the partner system is currently not available.

Measure:
Repeat the command later. If the error persists, contact the system or network administrator.

109 Request <Request id>. Connection setup rejected by local transport system

110 Request <Request id>. Data integrity check indicates an error

Meaning:
The integrity of the data was violated.

111 Encryption/data integrity check not possible. Encryption switched off

Meaning:
There is no key pair set or the key length was set to 0. Requests can only be carried out without data encryption or a data integrity check.

Measure:
Repeat the request without data encryption, create a key or set a key length >0.

112 Request <Request id>. Data integrity check not supported by partner

Meaning:
The partner system does not support the data integrity check.

Measure:
Repeat the request without a data integrity check.

113 Request <Request id>. User data encryption not possible for this request

Meaning:
The partner system does not support the data encryption function.

Measure:
Repeat the request without data encryption or install openFT-CR (or have it installed) on the remote system.

114 Request <Request id>. Identification of local system rejected by remote system '<partner>'

Meaning:
For security reasons or because of an inconsistency, the partner did not accept the instance identification of the local system (for example, because in a network description file both the instance identification and migration identification %.processor.entity occur for different partners).

Measure:
Ensure that the local identification has been entered correctly on the partner system and has not been assigned to a different partner.

115 Request <Request id>. Interrupted by remote system

116 Local application '<1>' not defined

Meaning:
The local application is not defined in the transport system, or the tnsxd process is not running in the Unix system.

Measure:
Make the local application known to the local transport system, or start the tnsxd process.

117 Local application '<1>' not available

118 Request <Request id>. Authentication of local system failed

Meaning:
The local system could not be authenticated by the partner system.

Measure:
Give the current public key file to the partner and name it correctly there. Repeat the command.

119 Request <Request id>. Local system unknown in remote system

Meaning:
The local system is not known on the partner system (e.g. BS2000/OSD or z/OS).

Measure:
Make the local system known on the partner system and repeat the command.

120 Remote system '<partner>' unknown

Meaning:
The partner specified as the remote system cannot be expanded to an address on the local system.

Measure:
Correct the specification for the partner or add the partner to the partner list and repeat the command.

121 Request <Request id>. Authentication of partner failed

Meaning:
The remote system could not be authenticated by the local system.

Measure:
Get the current public key file from the partner and name it correctly.

122 Request <Request id>. FT session rejected or disconnected.
Reason <2>

123 Request <Request id>. OSS call error <2>

Meaning:
The command was not executed because the session instance detected a communication error.
<2>: error code.

Measure:
Take the appropriate action in accordance with the error code.

124 Request <Request id>. No free connection

Meaning:
No more transfers are possible because the maximum number of simultaneous transfers has been reached.

Measure:
Check whether the transport system is working (or have it checked).

125 Request <Request id>. Connection lost

Meaning:
No data transfer took place because of a line interrupt or a line protocol error.

Measure:
Repeat the request.

126 Request <Request id>. Transport system error. Error code <2>

Meaning:
An error occurred in the transport system during processing of a /START-FT command or ftstart or a file transfer or file management request.

Measure:
Take the appropriate action in accordance with the error code. Most often the occurrence of this message indicates that the partner addressed is not known to the transport system. Contact system administrator to make sure there is an entry for the partner system.

127 Request <Request id>. No data traffic within <2> seconds

Meaning:
No data transfer took place within the period of seconds specified because, for example, the connection is interrupted, the partner is not sending and the local system is waiting for data.

Measure:
Repeat the request.

140 Request <Request id>. Remote system: openFT is not authorized to execute requests for this user

141 Request <Request id>. Remote system: Directory '<remote file>' is not empty

Meaning:
The command could not be executed because there are files in the specified directory of the partner system.

Measure:
Delete all the files in the directory first and repeat the command.

142 Request <Request id>. Remote system: File attributes do not match the request parameters

Meaning:
The command could not be executed because the file attributes on the remote system do not agree with the request parameters (e.g. a directory was specified instead of a remote file).

Measure:
Check the file name on the remote system and correct it.
Repeat the command.

143 Request <Request id>. Remote system: Attributes could not be modified

Meaning:
The properties of the file could not be modified as desired in the command. Possible reasons are:

For the remote file:
- No access rights to the file.
- The combination of access rights required is not supported by the remote system.
- If the remote system is a BS2000: the file is protected by ACL.

144 Request <Request id>. Remote system: File/directory '<remote file>' could not be created

Meaning:
The command was not executed because the file owner and user requesting the creation of a receive file are not the same.

Measure:
Match the user ID in the receive system's transfer admission to the ID of the receive file owner.
Repeat the command.

145 Request <Request id>. Remote system: CCS name unknown or not supported

Meaning:
The request could not be carried out because the CCS names of the send and receive files cannot be mapped to each other or because the partner system does not support the receipt of files in a transparent format.

146 Request <Request id>. Remote system: Higher-level directory not found

Meaning:
The command was not executed because the higher-level directory could not be found on the partner system.

Measure:
Create the directory on the remote system or correct the remote directory name and repeat the command.

147 Request <Request id>. Remote system: File/directory '<remote file>' already exists

Meaning:
The command was not executed. Possible reasons:
- The command was not executed because an existing receive file cannot be created with the *-n* option. *-n* may also have been set by a restriction in the access authorization used.
- *ftcredir*: The specified directory already exists.

Measure:
Either delete the receive file before repeating the command or reenter the command specifying option *-o* or using different access authorization.

148 Request <Request id>. Remote system: Transfer of file generation groups not supported

Meaning:
The command was not executed because the FT system can only transfer single file generations.

Measure:
Repeat the command using the name of a single file generation.

149 Request <Request id>. Remote system: Access error for ’<remote file>’ <3>

Meaning:
<3>: DMS error, possibly the transfer ID
The FT system continues to run after output of the message.

Measure:
Take the appropriate action in accordance with the error code.

150 Request <Request id>. Remote system: Resulting file name too long

Meaning:
A syntax error other than 'Mandatory parameter missing' (703) or 'keyword unknown' has been detected.
Possible reasons:
– Values assigned outside the valid range
– Invalid operand separators
– Invalid value assignment characters
– Partially qualified file names

Measure:
Repeat the command using the correct syntax.

151 Request <Request id>. Remote system: File locked to prevent multiple access

Meaning:
The command was not executed because either the send file or the receive file is already locked by another process to prevent it from being updated simultaneously.

Measure:
Repeat the command later or unlock the file on the remote system. After a system crash in BS2000 you may need to call VERIFY for files not closed correctly.

If the lock is caused by an FT request, it will be released automatically when the request is finished.

152 Request <Request id>. Remote system: No file or directory name specified

Meaning:
The command was not executed because the file ID was neither specified explicitly nor by the transfer admission used.

Measure:
Repeat the command, specifying the file ID explicitly or using a transfer admission that defines the file ID.

153 Request <Request id>. Remote system: Invalid management password

154 Request <Request id>. Remote system: File/directory '<remote file>' not available

Meaning:
The command was not executed because the volume for either the send file or the receive file is not mounted, unknown or reserved, the file extends over more than one private disk, or an attempt has been made to transfer a file migrated by HSMS.

Measure:
Inform the operator if necessary or carry out an HSMS recall for the file. Repeat the command.

155 Request <Request id>. Remote system: File/directory '<remote file>' not found

Meaning:
The command was not executed because the send file is not or no longer in the catalog or on a volume of the remote system.

Measure:
Correct the remote file name, read the file in from tape or restore the send file. Repeat the command.

156 Request <Request id>. Remote system: Home directory not found

157 Request <Request id>. Remote system: Renaming not possible

158 Request <Request id>. Remote system: Not enough space for '<remote file>'

Meaning:
The command was not executed (any further) because
– the permissible storage space on the receive system for the user ID specified in transfer admission has been used up, or
– the send file contains too long a sequence of empty blocks, or
– the primary and/or secondary allocation of the password-protected receive file is too small.

The receive file is no longer created/extended after the problem has occurred.

Measure:
Take the appropriate action depending on the cause of the error:
– delete all files no longer required on the receive system, or
– ask the system administrator to allocate more storage space, or
– remove empty blocks from the send file, or
– increase the receive file's primary/secondary allocation.

If option *-e* is specified, restore the receive file.
Repeat the command.

159 Request <Request id>. Remote system: File owner unknown

Meaning:
The command was not executed because the owner of either the send file or the receive file was not defined on the relevant system or because the file owner and the user requesting the creation of a receive file are not the same.

Measure:
Define the file owner, correct transfer admission or file name.
Repeat the command.

160 Request <Request id>. Remote system: Invalid file password

Meaning:
The command was not executed because the password for the send file or the receive file is missing or incorrect.

Measure:
Correct the password in the file description or the command.
Repeat the command.

161 Request <Request id>. Remote system: Retention period of file not yet expired

Meaning:
The command was not executed because the retention period protecting the receive file against overwriting has not yet expired.

Measure:
Correct the transfer direction, retention period or file name.
Repeat the command.

162 Request <Request id>. Remote system: File/directory '<remote file>' is read only

Meaning:
The file or directory is write-protected.

Measure:
Correct the remote file name or remove the write protection of the remote file. Repeat the command.

163 Request <Request id>. Remote system: File structure not supported

Meaning:
The request cannot be carried out because the file structure is not supported. For example, an attempt was made to get a PLAM library or ISAM file from the BS2000 system.

Measure:
Transfer the file transparently.

164 Request <Request id>. Remote system: Syntax error in resulting file name

Meaning:
A syntax error other than 'Mandatory parameter missing' (703) or 'keyword unknown' has been detected.

Possible reasons:
- Values assigned outside the valid range
- Invalid operand separators
- Invalid value assignment characters
- Partially qualified file names

Measure:
Repeat the command using the correct syntax.

165 Request <Request id>. Remote system: Transparent file transfer not supported

Meaning:
The request could not be carried out because the partner system does not support the transfer of files in a transparent format.

166 Request <Request id>. Remote system: Extension of file not possible for transparent transfer

Meaning:
The command could not be executed because it is not possible to add to a file in a transparent transfer.

167 Request <Request id>. Remote system: Access to '<remote file>' denied

Meaning:
The command was not executed because the remote file only permits certain access modes.

Measure:
Correct the transfer direction, file name or file protection attributes on the remote system. Repeat the command.

168 Request <Request id>. Remote system: Follow-up processing exceeds length limit

Meaning:
The length of follow-up processing was exceeded; see the command syntax description.

Measure:
Shorten the follow-up processing, or use procedures.
Repeat the command.

169 Request <Request id>. Remote system: Transfer admission invalid

Meaning:
The command was not executed because the specifications in one of the transfer admission operands are incorrect or the request was rejected by FTAC because of insufficient authorization.

Measure:
Define the requisite transfer admission or correct it or check the authorization entered in FTAC. Repeat the command if necessary.

170 Request <Request id>. Remote system: Function not supported

171 Request <Request id>. Remote system: Processing admission invalid

172 Request <Request id>. Remote system: Request queue full

195 Request <Request id>. Remote system: openFT is no longer authorized to execute requests for this user

196 Request <Request id> has been canceled in the remote system

Meaning:
The request was deleted on the remote system before termination.

197 Request <Request id>. Remote system: File/directory '<remote file>' could not be created

Meaning:
The command was not executed because the file owner and user requesting the creation of a receive file are not the same.

Measure:
Match the user ID in the receive system's transfer admission to the ID of the receive file owner. Repeat the command.

198 Request <Request id>. Remote system: Higher-level directory no longer found

199 Request <Request id>. Remote system: I/O error for '<remote file>

Meaning:
An error occurred at input/output. Possible cause:
– BS2000: DMS error, possibly the transfer ID.
– The send or receive files was deleted during transfer.

The FT system continues to run after the message has been issued.

Measure:
Take the appropriate action in accordance with the error code.

200 Request <Request id>. Remote system: File now locked to prevent multiple access

Meaning:
The command was not executed because either the send file or the receive file is already locked by another process to prevent it from being updated simultaneously. An attempt is made, for example, to access a library opened in z/OS.

Measure:
Repeat the command later or unlock the file.
After a system crash you may need to verify files not closed correctly.
If a lock is caused by an FT request, it will be released automatically when the request is finished.

201 Request <Request id>. Remote system: File/directory '<remote file>' no longer available

Meaning:
The command was not executed because the volume for either the send file or the receive file is not mounted, unknown or reserved, or because the file extends over more than one private disk or an attempt has been made to transfer a file migrated by HSMS.

Measure:
Inform the operator if necessary or carry out an HSMS recall for the file. Repeat the command.

202 Request <Request id>. Remote system: File/directory '<remote file>' no longer found

Meaning:
The command was not executed because the remote file is not or no longer in the catalog or on a volume of the corresponding system (e.g. after a restart).

Measure:
Restore the remote file. Repeat the command.

203 Request <Request id>. Remote system: Home directory no longer found

204 Request <Request id>. Remote system: File/directory '<remote file>' gets no more space

Meaning:
The command was not executed (any further) because
- the permissible storage space on the receive system for the user ID specified in transfer admission has been used up, or
- the send file contains too long a sequence of empty blocks, or
- the primary and/or secondary allocation of the password-protected receive file is too small.

The receive file can no longer be created/extended after the problem occurs.

Measure:
Take the appropriate action depending on the cause of the error:
– delete all files no longer required on the receive system, or
– ask the system administrator to allocate more storage space, or
– remove empty blocks from the send file, or
– increase the receive file's primary/secondary allocation.

If option *-e* is specified, restore the receive file. Repeat the command.

205 Request <Request id>. Remote system: File owner no longer known

Meaning:
The command was not executed because the owner of either the send file or the receive file is not defined on the relevant system, or because the file owner and the user requesting the creation of the receive file are not the same.

Measure:
Define the file owner, correct transfer admission or file name.
Repeat the command.

206 Request <Request id>. Remote system: Pre-/post-processing error

Meaning:
The command executed in local pre-/postprocessing returned an exit code other than 0.

Measure:
Correct the pre-/post-processing command and issue it again.

207 Request <Request id>. Remote system: Exit code <2> during pre-/post-processing

Meaning:
The command executed in local pre-/postprocessing returned the exit code <2>.

Measure:
Correct the pre-/post-processing command in accordance with the exit code and issue it again.

208 Request <Request id>. Remote system: File password no longer valid

Meaning:
The command was not executed because the password for the send file or receive file is missing or incorrect.

Measure:
Correct the password in the file description or the command.
Repeat the command.

209 Request <Request id>. Remote system: File/directory '<remote file>' is now read only

210 Request <Request id>. Remote system: File structure error

Meaning:
The command was not executed due to a file structure error.
File structure errors include:
- The attributes of the send file are incomplete.
- The data of the send file is incompatible with its structure attributes.
- The records of the send file are too long.
- If the *-e* option is specified, the send file and receive file have different structures (e.g. fixed-/variable-length records).
- BS2000: The send or receive file is a member of an old LMS library (not PLAM).
- BS2000: The send file has an odd block factor (e.g. BLKSIZE= (STD,1)), and the receive file is stored on an NK4 pubset.

Measure:
Correct the file or file attributes. If *-e* option is specified, restore the receive file. Repeat the command.

211 Request <Request id>. Remote system: NDMS error <2>

Meaning:
Repeat the request a little later.

212 Request <Request id>. Recovery failed

Meaning:
The restart could not be carried out. It may not have been possible to complete restart-capable pre-/post-processing before termination of the server process (waiting time: max. minutes).

Measure:
Repeat the command.

213 Request <Request id>. Remote system: Resource bottleneck

Meaning:
The order was rejected because the partner system currently does not have the resources available to accept requests.

Measure:
Repeat the request a little later.

214 Request <Request id>. Remote system: Access to '<remote file>' is no longer permissible

215 Request <Request id>. FTAM error <2>

216 Request <Request id>. Remote system: File structure not supported

217 Request <Request id>. Remote system: Retention period of file not yet expired

218 Request <Request id>. Remote system: Extension of file not possible for transparent transfer

9.1.3 Messages for administration commands and measurement data recording

In the case of the messages listed below, the value for *fthelp* must be increased by 1000, e.g. 1034 instead of 34.

20 openFT already started

Meaning:
openFT can only be started once in each instance.

Measure:
Terminate openFT if necessary.

21 Request must be canceled without FORCE option first

Meaning:
Before the FORCE option is used, the command must be called without the FORCE option.

Measure:
Issue the command without the FORCE option first.

29 Maximum number of key pairs exceeded

Measure:
Before a new key pair set can be created, an older key pair set must be deleted.

30 Warning: last key pair deleted

Meaning:
The last key pair set has been deleted. Without a key pair set, encrypted transfer, authentication and data integrity checking are not possible.

Measure:
Create a new key pair set.

31 No key pair available

Meaning:
All transfers are carried out without encryption.

Measure:
Create a new key pair set, if necessary.

32 Last key pair must not be deleted

33 The public key files could not be updated

Meaning:
The contents of the *syspkf* file could not be fully updated.
Possible reasons:
– The *syspkf* file is locked.
– There is not enough disk space to allow the file to be created.

Measure:
Take the appropriate action depending on the cause of the error:
– Unlock the file.
– Allocate disk space or have your system administrator do it.

Update the key with *ftupdk*.

34 Command only permissible for FT, FTAC or ADM administrator

Meaning:
Only the FT, FTAC or ADM administrator is permitted to use the command.

Measure:
Have the command executed by the FT, FTAC or ADM administrator.

35 Command only permissible for FT administrator

Meaning:
Only the FT administrator is permitted to use the command.

Measure:
Have the command executed by the FT administrator.

36 User not authorized for other user Ids

Meaning:
The user is not authorized to use a different user ID in the command.

Measure:
Specify your own ID, or have the command executed by the FT or FTAC administrator.

37 Key reference unknown

Meaning:
The specified key reference is unknown.

Measure:
Repeat the command with an existing key reference.

38 Request <Request id> is in the termination phase and can no longer be canceled

39 openFT not active

Meaning:
openFT is not started.

Measure:
Start openFT, if necessary.

40 Config user ID unknown or not enough space

Meaning:
The Config user ID of the current instance is unknown or the disk space allocated is insufficient to allow creation of the request file, the file for storing trace data, or the key files.

Measure:
Either create the Config user ID or increase its disk space allocation or have your system administrator do it.

41 Specified file is not a valid trace file

42 openFT could not be started

43 Partner with same attribute <attribute> already exists in partner list

Meaning:
There is already a partner entry with the same attribute <attribute> in the partner list.

Measure:
The attribute <attribute> in partner entries must be unique. Correct the command accordingly and try again.

44 Maximum number of partners exceeded

Meaning:
The partner list already contains the maximum permissible number of partner entries.

Measure:
Delete partners that are no longer required.

45 No partner found in partner list

Meaning:
A partner for the specified selection could not be found in the partner list.

Measure:
Check if the specified partner name or address was correct.
If necessary, repeat the command using the correct name or address.

46 Modification of partner protocol type not possible

Meaning:
The protocol type of the partner entry cannot be changed subsequently.

Measure:
Delete the partner from the partner list, if necessary, and enter it again with a new protocol type.

47 Request <Request id> not found

Meaning:
The request with the transfer ID <Request id> could not be found.

Measure:
Specify the existing transfer ID and repeat the command.

48 Active requests could not yet be deleted

49 CCS name '<1>' unknown

50 ftscript process could not be started

51 Error displaying an ftscript user

52 ftscript user number limit exceeded

53 ftscript chapter not found

54 ftscript id not found

55 ftscript file not found

56 ftscript request is still running

57 Inbound requests cannot be modified

58 The ADM trap server configuration is invalid

59 monitoring is not active

Meaning:
The command is only supported if monitoring is activated.

Measure:
Activate monitoring in the operating parameters and repeat the command.

60 File could not be created

Meaning:
The command was not executed because the local file could not be created.

Measure:
Check the directory and access rights. Repeat the command.

61 Higher-level directory not found

Meaning:
The local file could not be created when exporting the configuration data because the specified path does not exist.

Measure:
Create or correct the path for the configuration file and repeat the command.

62 File already exists

Meaning:
The command was not executed because the specified file already exists.

Measure:
Either delete the existing configuration file or choose a different name and repeat the command.

63 Resulting file name too long

Meaning:
The filename has the wrong syntax or is too long. Specifying a partially qualified filename may be the cause of the error.

Measure:
Repeat the command using the correct syntax.

64 File locked to prevent multiple access

Meaning:
The command was not executed because the file is already locked by another process.

Measure:
Repeat the command later.

65 File not found

Meaning:
The command was not executed because the specified file was not found.

Measure:
Correct the file name and repeat the command.

66 Not enough space for file

Meaning:
The command was not executed because the permitted storage space on the local volume is exhausted.

Measure:
Take appropriate measures depending on the cause of the error.

- Delete any files that are no longer required or
- Request the system administrator to assign more storage space.

67 Syntax error in resulting file name

Meaning:
The file cannot be accessed because the absolute file name has become too long, for instance.

Measure:
Shorten the path or the file name. Repeat the command.

68 Access to file denied<2>

Meaning:
The command was not executed because the file only permits certain access modes (e.g. read-only).

Measure:
Correct the file name or the file protection attributes.
Repeat the command.

69 Error accessing file<2>

Meaning:
<2>: DMS error

Measure:
Take appropriate measures depending on the error code.

70 Configuration data invalid

Meaning:
The configuration data is syntactically or semantically incorrect and can therefore not be imported.

Measure:
Correct the error on the basis of the additional diagnostic output and then repeat import of the configuration data.

71 Import of configuration data not possible while remote administration server is started

Meaning:
The changes to the configuration data are so extensive that they can only be imported when the remote administration server has been terminated.

Measure:
Terminate openFT using the *ftstop* command and then attempt to import the configuration data again.

73 Command aborted

Meaning:
The user has cancelled the command.

74 Command only permissible for ADM administrator on a remote administration server

Meaning:
The command is only permitted for the ADM administrator.

Measure:
Have the ADM administrator execute the command if necessary.

9.2 FTAC messages

001 FTAC version $VERSION active

003 $NUMBER logging records deleted

050 Lower ADM-level remains in effect

051 Transfer admission exists as user ID

052 Information incomplete

053 No FT profile found

054 No information available

055 Partner restriction does no longer exist

056 Transfer admission locked

057 Attributes of transfer admission are ignored

070 Shortage of resources

071 openFT not active

100 FT profile already exists

101 Transfer admission already exists

102 File already exists

103 Invalid file content or access to file denied

104 Access to directory denied

105 Access to file denied

106 Access to temporary file denied

107 No space available

108 The version of export file is not compatible with current version

109 File is no FTAC export file

Meaning:
A *ftshwe* or *ftimpe* command was issued for a file which is not a FTAC backup file.

110 File name too long

111 Syntax error in file name

112 Expiration date not valid

150 User not authorized for FTAC commands

151 User not authorized for this modification

152 User not authorized for other user IDs

153 User not authorized for other owner IDs

154 No authorization for deletion of log records

155 User not authorized for diagnose

156 Command allowed for FTAC administrator only

157 No authorization for this set of parameters

170 Given partner unknown

171 Given FT profile name unknown

172 Invalid user admission

173 Invalid processing admission

174 Modification invalid for not unique selection criteria

175 Modification invalid for standard authorization record

176 Given user ID unknown

177 File unknown

178 Multiple partner specified

179 Violation of maximal number of partners

180 Multiple user ID specified

181 Multiple FT profile name specified

182 Total maximum partner length exceeded

183 Partner not supported

184 Transfer admission of standard profile must be @n

185 Combination of these transfer functions not allowed

200 Follow-up processing too long

201 User ID too long

202 Profile name too long

203 Transfer admission too long

204 Partner too long

205 Fully qualified file name too long

Meaning:
By extension with absolute path name, the maximum value of 512 characters was exceeded.

206 Partially qualified file name too long

207 Processing command too long

208 Invalid date specified

209 Invalid time specified

210 Transfer admission too short

211 Parameters $PAR1 and $PAR2 must not be specified together

212 License check error $NUMBER for FTAC

213 Mandatory parameter profile name is missing

214 Mandatory parameter file name is missing

215 Syntax error in parameter $PARAMETER

216 Password too long

217 Text too long

218 Too many partners

219 Too many users

220 Too many profiles

250 Initialization of FTAC failed

251 FTAC not available

252 FTAC version incompatible

253 FTAC command not found in syntaxfile

254 System error. Errorcode $NUMBER

255 System error

If message 254 or 255 is displayed, please follow the instructions given in the chapter “What if ...” on page 375.

10 Appendix

This chapter lists the commands in the tool command library, describes the samples delivered with openFT and the CSV outputs from the openFT commands. In addition, you find the description of the BSFT commands and the associated messages.

10.1 Tool Command Library

The following tool commands are supplied with openFT:

– ft_tar

– ft_gzip

– ft_b2u and ft_u2b

– ft_mget

ft_tar and ft_gzip are the Gnu tar and Gnu zip tools subject to the Gnu Public License (GPL). These tools are supplied with openFT but are not subject to the openFT license, which means that you can copy and distribute them as long as you abide by the GPL. Fujitsu Technology Solutions reserves the right to stop supplying these tools in following versions or corrections versions of openFT or to supply them although they are not fully compatible with these versions. Renaming the tools to *ft_tar* and *ft_gzip* serves only to prevent collisions of installations on the various platforms.

An openFT user can therefore use these functions in procedures, preprocessing, post-processing or follow-up processing with a defined scope of functions. You can call up a short description of the functionality available using the "--help" option. You should only use the subset of functions described below if possible to minimize the possibility of encountering incompatibilities in later versions.

10.1.1 ft_tar

GNU 'tar' saves many files together into a single tape or disk archive, and can restore individual files from the archive.

Usage

ft_tar [OPTION]... [FILE]...

If a long option shows an argument as mandatory, then it is mandatory for the equivalent short option also. Similarly for optional arguments.

Main operation mode:

-t, --list list the contents of an archive

-x, --extract, --get extract files from an archive

-c, --create create a new archive

-r, --append append files to the end of an archive

-u, --update only append files newer than copy in archive

Operation modifiers:

-k, --keep-old-files don't overwrite existing files when extracting

-U, --unlink-first remove each file prior to extracting over it

--recursive-unlink empty hierarchies prior to extracting directory

-O, --to-stdout extract files to standard output

Device selection and switching:

-f, --file=ARCHIVE use archive file or device ARCHIVE

Archive format selection:

-z, --gzip, --ungzip filter the archive through gzip

Informative output:

--help print this help, then exit

--version print tar program version number, then exit

-v, --verbose verbosely list files processed

FILE may be a file or a device.

This `tar' defaults to `-f- -b20'.

Report bugs to <tar-bugs@gnu.org>.

10.1.2 ft_gzip

Usage

ft_gzip [-OPTION] [file ...]

-c --stdout write on standard output, keep original files unchanged

-d --decompress decompress

file... files to (de)compress. If none given, use standard input.

10.1.3 ft_b2u and ft_u2b

These two commands are used to convert data between binary format and user format (record length fields).

- The *ft_b2u* command converts files with binary data into files which contain data in user format (record length fields)
  (Tcl/Tk script to convert binary byte stream to openFT user format).
- The *ft_u2b* command converts files containing data in user format (record length data) into files in binary format
  (Tcl/Tk script to convert openFT user format to binary byte stream).

Format

ft_b2u -r=<1...32000> [-rf=1...32000>] [-rl=<1...32000>]

ft_u2b <inputfile> [<outputfile>]

Description

-r Length of the records into which the byte stream is to be converted.

-rf Optional: Length of the first record.

-rl Optional: Length of the last record.

inputfile
: Name of the file in user format or '-' (hyphen) for *stdin*.

outputfile
: Name of the binary file.

  Default value: *stdout*

10.1.4 ft_mget - Fetching multiple files

ft_mget allows you to fetch synchronously or asynchronously multiple files from a remote partner computer. You specify the files using wildcards. To do this, *ft_mget* uses the *ncopy* (synchronous) or the *ft* (asynchronous) command internally. The transfer mode (synchronous or asynchronous) is controlled via the *-async* option.

Format

```
ft_mget -h |
        [ -async ]
        [ -t | -u | -b ][ -x ]
        [ -o | -e | -n ]
        [ -k | -z ][ -c ][ -S | -s ][ -m=n | -m=f | -m=a ]
        <partner 1..200>!<file name with wildcard 1..512>
        <prefix 0..511>%
        <transfer admission 8..67> | @n |
            <user ID 1..67>[,[<account 1..64>][,<password 1..64>] ]
        [ -p=[<password 1..64>] ][ -di ]
        [ -lc=<CCS name 1..8> ][ -rc=<CCS name 1..8> ]
        [ -ls=<follow-up proc 1..1000> ][ -lf=<follow-up proc 1..1000> ]
        [ -rs=<follow-up proc 1..1000> ][ -rf=<follow-up proc 1..1000> ]
        [ -r=v[<1..32767>] | -r=f[<1..32767>] |
          -r=u[<1..32767>] | -r=<1..32767> ]
        [ -tff=b | -tff=s ][ -trf=u ]
        [ -av=i | -av=d ] [ -ac=<new account number 1..64> ]
        [ -am=[r][i][p][x][e][a][c][d] | -am=@rw | -am=@ro ]
        [ -lq=<legal qualification 1..80> ]
        [ -pr=n | -pr=l ]
        [ -sd=yyyymmdd | +<start date 0..ddddd> ]
        [ -st=[+]<start time hhmm> ]
        [ -cd=yyyymmdd | +<cancel date 0..ddddd> ]
        [ -ct=[+]<cancel time hhmm> ]
```

Description

Only the differences compared with the *ncopy* and *ft* command are described below. The other parameters have the same meanings as in the *ncopy* command (see page 331) and the *ft* command (see page 147).

Note that the same conditions apply to the *-c* option (encryption of user data) as for the *ft* or *ncopy* command, i.e. openFT-Crypt must be installed and the partner system must support encryption.

-async

The files are fetched asynchronously. In this event, you must not specify the *-s* option. All other parameters are permitted.

In the case of asynchronous transfer, the number of transfer requests that can be processed simultaneously is limited by the size of the request queue. If you wish to fetch a large number of files asynchronously using ft_mget, the FT administrator may have to increase the maximum size of the request queue.

This is done using the ftmodo -*rql*= command or the openFT Explorer (*Administration* - *Operating Parameters*, *General* tab, *Maximum number*: *Requests* option). For further details, refer to the openFT manual "Installation and Administration".

-async not specified

If you omit -async, the files are fetched synchronously. In this event, you must not specify the following options:

- *-ls* and *-lf* (local follow-up processing)
- *-pr* (priority)
- *-sd* and *-st* (start date and time)
- *-cd* and *-ct* (deletion date and time)

All other parameters are permitted.

transfer-admission | **@n** | userid[,[account][,password]]

Specification of the transfer admission is mandatory. Blanking of your entry is not supported. You are therefore not permitted to specify either the value *@d* or a user ID without password in the form *userid*,[*account*].

filename with wildcard

Specifies which files are to be fetched from the remote system.

You can only use wildcard characters in the final part of the name following the last slash (/) or backslash (\), not in the directory name. File names are case-sensitive with Unix and POSIX systems. Other partner systems are not case-sensitive. A BS2000 partner is regarded as a POSIX system if the specified file name starts with a POSIX pathname (i.e. with / or ./).

All files that match the pattern specified under *file name with wildcard* are transferred to the local computer synchronously by *ft_mget* in a loop of *ncopy* commands.

The following characters can be used to define a wildcard pattern:

* as a wildcard for any string (including an empty string).

? as a wildcard for any single character.

[chars]
as a wildcard for a single character from the set specified by *chars*. In *chars*, you can list individual characters or specify one or more character ranges in the form a-z. This selects all characters a through z (inclusive).

Example:
[aeiX-Z] stands for one of the characters a e i X Y Z.

\x as a wildcard for any single character. The backslash is used to cancel the special meaning of the characters *?[]\ in the specified wildcard pattern.

On Unix systems, steps must be taken to ensure that wildcard characters and the exclamation mark (!) are not interpreted or resolved by the local shell. For this reason, we strongly recommend that you enclose the expression *<partner 1..200>!<file name with wildcard 1..512>* in quotes, i.e. enter it in the form *'<partner 1..200>!<file name with wildcard 1..512>'*, e.g. `ft_mget 'server01!*.pdf'` ....

prefix%
Determines the names of the receive files in the local system.

You can specify %, %BASENAME, prefix%, or prefix%BASENAME:

% or %BASENAME
Each of these are replaced by the last part of the name of the remote file. The last part of the name starts after the last slash (/) or backslash (\) or a corresponding character in the remote system.

prefix% or prefix%BASENAME
You can also specify an optional prefix, e.g. *saved*.%BASENAME.

This prefix must end with a dot (.), a slash (/) or a backslash (\). The prefix can also contain the absolute or relative path of a directory that exists on the local computer. If the specified directory does not exist, *ft_mget* is not executed.

Note that the resulting file name must comply with the rules of the local system, otherwise the files will not be transferred.

Result messages and return codes

On success, *ft_mget* issues one of the following messages:

`<n> files successfully transferred` (synchronous transfer)

`Transfer of <n> files successfully initiated` (asynchronous transfer)

Where <*n*> stands for the number of files transferred synchronously or the number of asynchronous file transfer requests initiated. If no files that match the specified pattern were found on the remote system, the following message appears instead:

`No files corresponding to specified pattern found`

ft_mget normally terminates with the return code 0. If an error occurs during execution, the command terminates and returns one of the following return codes (*RC*):

RC	Output to stderr	Meaning
1	Invalid source parameter '<par>'. Source expected as <partner 1..200>!<file name with wildcard 1..512>.	The specification of the parameter used to specify the files to be transferred does not match the required format.
1	ft_mget syntax help	One of the mandatory parameters for *ft_mget* was not specified.
1	Invalid transfer admission specified.	*@d* or *userid*,[*account*], was specified in place of a transfer admission.
1	Parameter(s) '<par>' only allowed together with '-async'	The parameters <par> are only allowed for asynchronous file transfer.

RC	Output to stderr	Meaning
1	Parameter(s) '<par>' must not be specified together with '-async'	The parameters <par> are not allowed for asynchronous file transfer.
2	Given target directory '<dir>' does not exist.	The target directory specified does not exist on the local system.
3	Given target path must contain %, %BASENAME, or %FILENAME.	The parameter specified for the target of *ft_mget* does not end with one of the specified placeholders.
4	openFtCmd <ftshw> failed	The openFT command *ftshw* for determining the files in the specified remote directory failed.
5	ft::isAbort after openFtCmd <ftshw>	The openFT command *ftshw* for determining the files in the specified remote directory failed.
6	Remote directory <dir> on host <partner> could not be accessed (return code='<rc>', exit code= '<code>').	It is not possible to access the specified directory on the remote partner system.
6	Reading content of remote directory <dir> on host <partner> failed (return code='<rc>', exit code= '<code>').	It was not possible to read the specified directory on the remote partner system.
7	Not all files successfully transferred	At least one source file could not be transferred to the local system. The previous message(s) indicate(s) the file(s) concerned: Transfer of file '<file>' failed. return code is: '<rc>'

Example

You want to fetch synchronously all files on the Unix computer *MCH0001X* located in the directory *tmp/config* and whose names start with *cfg* onto the local computer and store them there in the *config* subdirectory of the current directory. The command is as follows:

```
ft_mget  'MCH0001X!/tmp/config/cfg*'
          config/copy.%BASENAME  mytad001
```

If, for instance, the source directory contains the files *cfg001*, *cfg002* and *cfg003*, *ft_mget* creates the local receive files *config/copy.cfg001*, *config/copy.cfg002* and *config/copy.cfg003*.
mytad001 is a valid FTAC transfer admission for the computer *MCH0001X*.

10.2 Sample files

openFT is supplied with a range of sample files that you can use for various purposes. Once openFT has been installed, you will find these files in the directory */opt/openFT/samples*.

ftadm

The file *config.xml* contains a simple sample configuration for remote administration. You can use this sample as a template and adapt it according to your needs.

ftscript

This directory contains examples for the openFT-Script interface. You will find a description of the interface in the manual "openFT for Unix and Windows Systems - openFT-Script Interface".

filedist.ftsc
: Distribute files to several different partner systems.

transsuc.ftsc
: Transfer a file to a partner system with follow-up processing.

treecopy.ftsc
: Transfer a complete directory tree to a partner system.

ftaccnt.xlt

The Excel template demonstrates how to evaluate the CSV output format of the logging commands and how to use them in Excel for accounting purposes.

sample1.c, sample2.c, sample3.c

These examples illustrate various options for using the C programming interface of openFT. You will find a description of the examples in the manual "openFT for Unix and Windows Systems - Program Interface".

sample1.c
: Transfer a file asynchronously

sample2.c
: Transfer several files with follow-up processing.

sample3.c
: Show the contents of a remote directory.

Sample1.java, Sample2.java, Sample3.java

These examples illustrate the Java programming interface of openFT. How to compile and run the examples is described in the section “Programming with JAVA” on page 374.

Sample1.java
: Transfer a file asynchronously

Sample2.java
: Transfer several files with follow-up processing.

Sample3.java
: Show the contents of a remote directory.

pattern**treecopy-get, treecopy-send, treecopy-send-unique**

These examples illustrate various ways of transferring a complete directory to Unix or Windows partner systems.

treecopy-get
: Fetch all files of a directory from a partner system using preprocessing. In this example, preprocessing is used in the remote system without an intermediate file being specified.

treecopy-send
: Pack all files of a directory in a tar archive using preprocessing, transfer them to a partner system and unpack them there using postprocessing.

treecopy-send-unique
: Pack all files of a directory in a tar archive using preprocessing, transfer them to a partner system and unpack them there using follow-up processing.

  The use of %UNIQUE in the receive file name allows several scripts to be executed concurrently.

10.3 Structure of CSV Outputs

The CSV output of the commands *ftshwact* and *ftshws* is described in the manual „openFT-Script Interface“

10.3.1 ftshw

The following table indicates the CSV output format for file attributes.

Column	Type	Values
FileName	String	Value enclosed in double quotes / *NSPEC
StorageAccount	String	Value enclosed in double quotes / *NSPEC
CreIdentity	String	Value enclosed in double quotes / *NSPEC
CreTime	yyyy-mm-dd hh:mm:ss	Value / *NSPEC
ModIdentity	String	Value enclosed in double quotes / *NSPEC
ModTime	yyyy-mm-dd hh:mm:ss	Value / *NSPEC
ReaIdentity	String	Value enclosed in double quotes / *NSPEC
ReaTime	yyyy-mm-dd hh:mm:ss	Value / *NSPEC
AtmIdentity	String	Value enclosed in double quotes / *NSPEC
AtmTime	yyyy-mm-dd hh:mm:ss	Value / *NSPEC
FileType	String	*BIN / *DIR / *TEXT / *NONE / *NSPEC
CharSet	String	*VISIBLE / *IA5 / *GRAPHIC / *GENERAL / *NONE / *NSPEC
RecFormat	String	*VAR / *FIX / *NSIG / *NSPEC
RecSize	Number	Value / *NSPEC
FileAvail	String	*IMMEDIATE / *DEFERRED / *NSPEC

Column	Type	Values
AccessRights	String	Value / *NSPEC
FileSize	Number	Value / *NSPEC
MaxFileSize	Number	Value / *NSPEC
LegalQualif	String	Value enclosed in double quotes / *NSPEC
CcsName	String	Value / *NSPEC

Example

```
$ ftshw bs2partn!aaa.e42 transbs2 -CST
   FileName;StorageAccount;CreIdentity;CreTime;ModIdentity;
   ModTime;ReaIdentity;ReaTime;AtmIdentity;AtmTime;FileType;
   CharSet;RecFormat;RecSize;FileAvail;AccessRights;FileSize;
   MaxFileSize;LegalQualif;CcsName
   "aaa.e42";*NSPEC;"maier";*NSPEC;*NSPEC;2008-03-17 13:01:34;
   *NSPEC;*NSPEC;*NSPEC;*NSPEC;*NSPEC;*NSPEC;*NSIG;*NSPEC;
   *NSPEC;r-pxeacd---;174;*NSPEC;*NSPEC;*NSPEC
```

10.3.2 ftshwa

The following table indicates the CSV output format of an admission set.

Column	Type	Values
UserId	String	Value enclosed in double quotes
UserMaxObs	Number	Value
UserMaxObsStd	String	*YES / *NO
UserMaxObr	Number	Value
UserMaxObrStd	String	*YES / *NO
UserMaxIbs	Number	Value
UserMaxIbsStd	String	*YES / *NO
UserMaxIbr	Number	Value
UserMaxIbrStd	String	*YES / *NO
UserMaxIbp	Number	Value
UserMaxIbpStd	String	*YES / *NO
UserMaxIbf	Number	Value
UserMaxIbfStd	String	*YES / *NO
AdmMaxObs	Number	Value
AdmMaxObsStd	String	*YES / *NO
AdmMaxObr	Number	Value
AdmMaxObrStd	String	*YES / *NO
AdmMaxIbs	Number	Value
AdmMaxIbsStd	String	*YES / *NO
AdmMaxIbr	Number	Value
AdmMaxIbrStd	String	*YES / *NO
AdmMaxIbp	Number	Value
AdmMaxIbpStd	String	*YES / *NO
AdmMaxIbf	Number	Value
AdmMaxIbfStd	String	*YES / *NO

Column	Type	Values
Priv	String	*YES / *NO
Password	String	*YES / *NO
AdmPriv	String	*YES / *NO

10.3.3 ftshwl

The following table indicates the CSV output format of a log record.

A format template in Microsoft Excel format is present in the following file as an example of a possible evaluation procedure:

/opt/openFT/samples/ftaccnt.xlt

Column	Type	Values
LogId	Number	Value
ReasonCode	String	Value enclosed in double quotes to prevent interpretation as a number. FTAC Reason Codes are output as hexadecimal strings.
LogTime	yyyy-mm-dd hh:mm:ss	Value
InitUserId	String	Value enclosed in double quotes / *REM
InitTsn	String	Value enclosed in double quotes / *NONE
PartnerName	String	Value enclosed in double quotes
TransDir	String	*TO / *FROM / *NSPEC
RecType	String	*FT / *FTAC / *ADM
Func	String	*TRANS-FILE / *READ-FILE-ATTR / *DEL-FILE / *CRE-FILE / *MOD-FILE-ATTR / *READ-DIR / *MOVE-FILE / *CRE-FILE-DIR / *DEL-FILE-DIR / *LOGIN *MOD-FILE-DIR / *REM-ADMIN / *REM-ADMIN-ROUT
UserAdmisId	String	Value enclosed in double quotes
FileName	String	Value enclosed in double quotes

Column	Type	Values
Priv	String	*NO / *YES for FTAC log records and entry of an FTAC profile; otherwise *NONE
ProfName	String	Value enclosed in double quotes / *NONE
ResultProcess	String	*NONE / *STARTED / *NOT-STARTED
StartTime	yyyy-mm-dd hh:mm:ss	Value
TransId	Number	Value
Write	String	*REPL / *EXT / *NEW / *NONE
StoreTime	yyyy-mm-dd hh:mm:ss	Value
ByteNum	Number	Value
DiagInf	String	Value enclosed in double quotes / *NONE
ErrInfo	String	Value enclosed in double quotes / *NONE
SecEncr	String	*YES or *NO
SecDichk	String	*YES or *NO
SecDencr	String	*YES or *NO
SecDdichk	String	*YES or *NO
SecLauth	String	*YES or *NO
SecRauth	String	*YES or *NO
RsaKeyLen	Number	Value, the space remains empty if SecEncr does not have the value *YES
SymEncrAlg	String	*DES / *AES-128 / *AES-256 Value, the space remains empty if SecEncr does not have the value *YES
CcsName	String	Value enclosed in double quotes

Column	Type	Values
AdminId	String	Value enclosed in double quotes
Routing	String	Value enclosed in double quotes
AdmCmd	String	Value enclosed in double quotes

10.3.4 ftshwm

The following table shows the CSV output format for the monitoring values for openFT operation if all the monitoring values are output (*ftshwm -csv @a*).

If the *-raw* option is specified, the duration values are not output (*Duxxx*, see footnote).

The default values are marked with "x" in the *Std* column. These are output if *ftshwm -csv* is specified without *@a* and without names being specified explicitly.

For a detailed description of the monitoring values, refer to the section "Description of the monitoring values" on page 292.

Name	Type	Value	Std
CurrTime	yyyy-mm-dd hh:mm:ss	current time	x
MonOn	yyyy-mm-dd hh:mm:ss	start time of measurement date recording or last change of configuration (a modification of PartnerSel/ReqSel has the same effect as a new start)	x
PartnerSel	String	*ALL / *NONE / OPENFT / FTAM / FTP	x
ReqSel	String	*ALL / ONLY-SYNC / ONLY-ASYNC / ONLY-LOCAL / ONLY-REMOTE	x
Data	String	FORM / RAW	x
ThNetbTtl	Number	Value	x
ThNetbSnd	Number	Value	x
ThNetbRcv	Number	Value	x
ThNetbTxt	Number	Value	
ThNetbBin	Number	Value	
ThDiskTtl	Number	Value	x
ThDiskSnd	Number	Value	x
ThDiskRcv	Number	Value	x
ThDiskTxt	Number	Value	
ThDiskBin	Number	Value	

Name	Type	Value	Std
ThRqto	Number	Value	x
ThRqft	Number	Value	
ThRqfm	Number	Value	
ThSuct	Number	Value	x
ThAbrt	Number	Value	x
ThIntr	Number	Value	x
ThUsrf	Number	Value	x
ThFoll	Number	Value	
ThCosu	Number	Value	
ThCofl	Number	Value	x
ThCobr	Number	Value	x
DuRqtlOut[1]	Number	Value	
DuRqtlInb[1]	Number	Value	
DuRqftOut[1]	Number	Value	
DuRqftInb[1]	Number	Value	
DuRqfmOut[1]	Number	Value	
DuRqfmInb[1]	Number	Value	
DuRqesOut[1]	Number	Value	
DuDnscOut[1]	Number	Value	
DuDnscInb[1]	Number	Value	
DuConnOut[1]	Number	Value	
DuOpenOut[1]	Number	Value	
DuOpenInb[1]	Number	Value	
DuClosOut[1]	Number	Value	
DuClosInb[1]	Number	Value	
DuUsrcOut[1]	Number	Value	
DuUsrcInb[1]	Number	Value	
StRqas	Number (100)[2]	Value	x
StRqaa	Number (100)[2]	Value	x

Name	Type	Value	Std
StRqwt	Number (100)[2]	Value	x
StRqhd	Number (100)[2]	Value	x
StRqsp	Number (100)[2]	Value	x
StRqlk	Number (100)[2]	Value	x
StRqfi	Number (100)[2]	Value	
StCLim	Number	Value	x
StCAct	Percent	Value	x
StRqLim	Number	Value	x
StRqAct	Percent	Value	x
StOftr	BOOL	1 (openFT protocol active) / 0 (not active)	x
StFtmr	BOOL	1 (FTAM protocol active) / 0 (not active)	x
StFtpr	BOOL	1 (FTP protocol active) / 0 (not active)	x
StTrcr	BOOL	1 (Trace switched on) / 0 (switched off)	

[1]is not output with option -raw
[2]number is the measured value multiplied by 100 (e.g. output 225 corresponds to value 2.25)

Examples

```
ftshwm -ty -csv @a
```

```
CurrTime;MonOn;PartnerSel;ReqSel;Data;ThNetbTtl;ThNetbSnd;ThNetb
Rcv;ThNetbTxt;ThNetbBin;ThDiskTtl;ThDiskSnd;ThDiskRcv;ThDiskTxt;
ThDiskBin;ThRqto;ThRqft;ThRqfm;ThSuct;ThAbrt;ThIntr;ThUsrf;ThFol
l;ThCosu;ThCofl;ThCobr;DuRqtlOut;DuRqtlInb;DuRqftOut;DuRqftInb;D
uRqfmOut;DuRqfmInb;DuRqesOut;DuDnscOut;DuDnscInb;DuConnOut;DuOpe
nOut;DuOpenInb;DuClosOut;DuClosInb;DuUsrcOut;DuUsrcInb;StRqas;St
Rqaa;StRqwt;StRqhd;StRqsp;StRqlk;StRqfi;StCLim;StCAct;StRqLim;St
RqAct;StOftr;StFtmr;StFtpr;StTrcr
*TIME;*TIME;*STRING;*STRING;*STRING;*INT;*INT;*INT;*INT;*INT;*IN
T;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*I
NT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT;*
INT;*INT;*INT;*INT;*INT;*INT;*INT;*INT(100);*INT(100);*INT(100);*INT(
100);*INT(100);*INT(100);*INT(100);*INT;*PERCENT;*INT;*PERCENT;*
BOOL;*BOOL;*BOOL;*BOOL
```

```
ftshwm -csv ThNetbTtl ThDiskTtl

CurrTime;MonOn;PartnerSel;ReqSel;Data;ThNetbTtl;ThDiskTtl
2008-02-28 15:40:01;2008-02-28 15:36:12;OPENFT,FTAM;ONLY-
ASYNC,ONLY-REMOTE;FORM;2681262;524064
```

10.3.5 ftshwo

The following table indicates the CSV output format of the operating parameters.

Column	Type	Values
PartnerLimit	Number	Value
ReqLim	Number	Value
TaskLim	Number	Value
ConnLim	Number	Value
ReqWaitLev	Number	Value
TransportUnitSize	Number	Value
PartnerCheck	String	*STD / *TRANSP-ADDR
SecLev	Number	*B-P-ATTR / Value
TraceOpenft	String	*STD / *OFF
TraceOut	String	*FILE / *OFF
TraceSession	String	*STD / *OFF
TraceFtam	String	*STD / *OFF
LogTransFile	String	*ON / *OFF
MaxInboundReq	Number	Value
MaxReqLifetime	String	Value / *UNLIMITED
SnmpTrapsSubsystemState	String	*OFF / *ON
SnmpTrapsFtState	String	*OFF / *ON
SnmpTrapsPartnerState	String	*OFF / *ON
SnmpTrapsPartnerUnreach	String	*OFF / *ON
SnmpTrapsReqQueueState	String	*OFF / *ON
SnmpTrapsTransSucc	String	*OFF / *ON
SnmpTrapsTransFail	String	*OFF / *ON
ConsoleTraps	String	*ON / *OFF
TeleService	String	empty [1]
HostName	String	Value / *NONE
Identification	String	Value enclosed in double quotes

Column	Type	Values
UseTns	String	*YES / *NO
ConsTrapsSubsystemState	String	*ON / *OFF
ConsTrapsFtState	String	*ON / *OFF
ConsTrapsPartnerState	String	*ON / *OFF
ConsTrapsPartnerUnreach	String	*ON / *OFF
ConsTrapsReqQueueState	String	*ON / *OFF
ConsTrapsTransSucc	String	*ON / *OFF
ConsTrapsTransFail	String	*ON / *OFF
FtLog	String	*ALL / *FAIL / *NONE
FtacLog	String	*ALL / *FAIL / *NONE
Trace	String	*ON / *OFF
TraceSelp	String	*ALL / *NONE / OPENFT / FTP / FTAM / ADM [2]
TraceSelr	String	*ALL / ONLY-SYNC / ONLY-ASYNC / ONLY-LOCAL / ONLY-REMOTE [2]
TraceOpt	String	*NONE / *BULK-DATA
KeyLen	Number	Value
CcsName	String	Value enclosed in double quotes
AppEntTitle	String	*YES / *NO
StatName	String	Value
SysName	String	Value
FtStarted	String	*YES / *NO
openftAppl	String	*STD / Value
ftamAppl	String	*STD / Value
FtpPort	Number	Value
ftpDPort	Number	Value / empty (internal function)
ftstdPort	String	Value / *STD
DynPartner	String	*ON / *OFF
ConTimeout	Number	Value (internal function)
ChkpTime	Number	Value (internal function)

Column	Type	Values
Monitoring	String	*ON / *OFF
MonSelp	String	*ALL / OPENFT / FTP / FTAM / empty [2]
MonSelr	String	*ALL / ONLY-SYNC / ONLY-ASYNC / ONLY-LOCAL / ONLY-REMOTE [2]
AdmTrapServer	String	Value / *NONE
AdmTrapsFtState	String	*ON / *OFF
AdmTrapsPartnerState	String	*ON / *OFF
AdmTrapsPartnerUnreach	String	*ON / *OFF
AdmTrapsReqQueueState	String	*ON / *OFF
AdmTrapsTransSucc	String	*ON / *OFF
AdmTrapsTransFail	String	*ON / *OFF
AdminConnLim	String	Value
AdmPort	String	Value / *NONE
OpenftApplState	String	*ACTIVE / *INACT / *DISABLED / *NAVAIL
FtamApplState	String	*ACTIVE / *INACT / *DISABLED / *NAVAIL
FtpState	String	*ACTIVE / *INACT / *DISABLED / *NAVAIL
AdmState	String	*ACTIVE / *INACT / *DISABLED
AdminLog	String	*ALL / *FAIL / *MODIFY / *NONE
CentralAdminServer	String	*YES / *NO
ActiveAppl	String	*ALL / *NONE / OPENFT / FTAM / FTP / ADM [2]

[1] Not relevant on Unix systems
[2] Combinations are also possible (not with *ALL and *NONE)

10.3.6 ftshwp

The following table indicates the CSV output format of an admission profile.

Column	Type	Values
ProfName	String	Value enclosed in double quotes
Priv	String	*YES / *NO
TransAdm	String	*NSPEC / *SECRET
Duplicated	String	*YES / *NO
LockedByImport	String	*YES / *NO
LockedByAdm	String	*YES / *NO
LockedByUser	String	*YES / *NO
Expired	String	*YES / *NO
ExpDate	yyyy-mm-dd	Value / *NRES
Usage	String	*PUBLIC / *PRIVATE / *NSPEC
IgnObs	String	*YES / *NO
IgnObr	String	*YES / *NO
IgnIbs	String	*YES / *NO
IgnIbr	String	*YES / *NO
IgnIbp	String	*YES / *NO
IgnIbf	String	*YES / *NO
Initiator	String	*LOC / *REM / *NRES
TransDir	String	*FROM / *TO / *NRES
MaxPartLev	Number	Value / *NRES
Partners	String	One or more FT partners, delimited by commas and enclosed in double quotes / *NRES
FileName	String	Value enclosed in double quotes / *NRES
Library	String	*YES / *NO / *NRES / Value enclosed in double quotes
FileNamePrefix	String	*YES / *NO
ElemName	String	Value enclosed in double quotes / *NRES / *NONE

Column	Type	Values
ElemPrefix	String	*YES / *NO
ElemVersion	String	Value enclosed in double quotes / *STD / *NONE / *NRES
ElemType	String	Value enclosed in double quotes / *NRES / *NONE
FilePass	String	*YES / *NRES / *NONE
Write	String	*NEW / *EXT / *REPL / *NRES
UserAdmId	String	Value enclosed in double quotes
UserAdmAcc	String	Value enclosed in double quotes / *NSPEC / *NRES
UserAdmPass	String	*OWN / *NSPEC / *NONE / *YES
ProcAdmId	String	Value enclosed in double quotes / *NRES / *SAME
ProcAdmAcc	String	Value enclosed in double quotes / *NRES / *SAME
ProcAdmPass	String	*NONE / *YES / *NRES /*SAME
SuccProc	String	Value enclosed in double quotes / *NONE / *NRES / *EXPANSION
SuccPrefix	String	Value enclosed in double quotes / *NONE
SuccSuffix	String	Value enclosed in double quotes / *NONE
FailProc	String	Value enclosed in double quotes / *NONE / *NRES / *EXPANSION
FailPrefix	String	Value enclosed in double quotes / *NONE
FailSuffix	String	Value enclosed in double quotes / *NONE
TransFile	String	*ALLOWED / *NOT-ALLOWED
ModFileAttr	String	*ALLOWED / *NOT-ALLOWED
ReadDir	String	*ALLOWED / *NOT-ALLOWED
FileProc	String	*ALLOWED / *NOT-ALLOWED
RemAdm	String	*ALLOWED / *NOT-ALLOWED
AccAdm	String	*ALLOWED / *NOT-ALLOWED
Text	String	Value enclosed in double quotes / *NONE

Column	Type	Values
DataEnc	String	*NRES / *YES / *NO
ModDate	yyyy-mm-dd hh:mm:ss	Value
AdmTrapLog	String	*ALLOWED / *NOT-ALLOWED

10.3.7 ftshwptn

The following table indicates the CSV output format of a partner.

Column	Type	Values
PartnerName	String	Value enclosed in double quotes
Sta	String	*ACT / *DEACT / *NOCON / *LUNK / *RUNK / *ADEAC / *AINACT / *LAUTH / *RAUTH / *NOKEY / *DIERR / *IDREJ
SecLev	String	*STD / *B-P-ATTR / Value enclosed in double quotes
Trace	String	*FTOPT / *STD / *ON / *OFF
Loc	Number	Value
Rem	Number	Value
Processor	String	Value enclosed in double quotes / empty
Entity	String	Value enclosed in double quotes / empty
NetworkAddr	String	Value enclosed in double quotes
Port	Integer	Value
PartnerCheck	String	*FTOPT / *STD / *TRANSP-ADDR / *AUTH / *AUTHM
TransportSel	String	Value enclosed in double quotes / empty
LastAccessDate	yyyy-mm-dd	Value
SessionSel	String	Value enclosed in double quotes / empty
PresentationSel	String	Value enclosed in double quotes / empty
Identification	String	Value enclosed in double quotes
SessRout	String	Value enclosed in double quotes / *ID / empty
PartnerAddr	String	Value enclosed in double quotes
Check	String	*FTOPT / *STD / *TRANSP-ADDR
AuthMand	String	*YES / *NO
Priority	String	*LOW / *NORM / *HIGH

10.3.8 ftshwr

The following table indicates the CSV output format of a request.

Short output is also possible with *ftshwr* (*ftshwr -s -csv*), see page 452.

Column	Type	Values
TransId	Number	Value
Initiator	String	*LOC / *REM
State	String	*LOCK / *WAIT / *HOLD / *FIN / *ACT / *CANC / *SUSP
PartnerName	String	Value enclosed in double quotes
PartnerState	String	Values
TransDir	String	*TO / *FROM
ByteNum	Number	Value / empty
LocFileName	String	Value enclosed in double quotes
LocElemName	String	empty
LocElemType	String	empty
LocElemVersion	String	empty
Prio	String	*NORM / *LOW
Compress	String	*NONE / *BYTE / *ZIP
DataEnc	String	*YES / *NO
DiCheck	String	*YES / *NO
Write	String	*REPL / *EXT / *NEW
StartTime	yyyy-mm-dd hh:mm:ss	Value
	String	*SOON
CancelTime	yyyy-mm-dd hh:mm:ss	Value
	String	*NO
Owner	String	Value enclosed in double quotes
DataType	String	*CHAR / *BIN / *USER
Transp	String	*YES / *NO

Column	Type	Values
LocTransAdmId	String	*NONE / Value enclosed in double quotes
LocTransAdmAcc	String	empty
LocProfile	String	*NONE / Value enclosed in double quotes / empty
LocProcAdmId	String	*NONE / Value enclosed in double quotes / empty
LocProcAdmAcc	String	empty
LocSuccProc	String	Value enclosed in double quotes / empty
LocFailProc	String	Value enclosed in double quotes / empty
LocListing	String	empty
LocMonjv	String	empty
LocCcsn	String	*STD / Value enclosed in double quotes
RemFileName	String	Value enclosed in double quotes / empty
RemElemName	String	Value enclosed in double quotes / empty
RemElemType	String	Value enclosed in double quotes / empty
RemElemVersion	String	Value enclosed in double quotes / empty
RemTransAdmId	String	Value enclosed in double quotes / empty
RemTransAdmAcc	String	Value enclosed in double quotes / empty
RemTransAdmAccount	String	Value enclosed in double quotes / empty
RemProfile	String	*YES / *NONE
RemProcAdmId	String	Value enclosed in double quotes / empty

Column	Type	Values
RemProcAdmAcc	String	empty
RemSuccProc	String	Value enclosed in double quotes / empty
RemFailProc	String	Value enclosed in double quotes / empty
RemCcsn	String	*STD / Value enclosed in double quotes
FileSize	Number	Value / empty
RecSize	Number	Value / empty
RecFormat	String	*STD / *VARIABLE / *FIX / *UNDEFINED
StoreTime	yyyy-mm-dd hh:mm:ss	Value
ExpEndTime	yyyy-mm-dd hh:mm:ss	Value / empty
TranspMode	String	*YES / *NO
DataEncrypt	String	*YES / *NO
TabExp	String	*AUTO / *YES / *NO
Mail	String	*ALL / *FAIL / *NO
DiagCode	String	Value / empty
FileAvail	String	*IMMEDIATE / *DEFERRED / *NSPEC
StorageAccount	String	Value / empty
AccessRights	String	Values / empty
LegalQualif	String	Value / empty
PartnerPrio	String	*LOW / *NORM / *HIGH
TargetFileForm	String	*STD / *BLOCK / *SEQ
TargetRecForm	String	*STD / *UNDEFINED
Protection	String	*STD / *SAME

Short output of ftshwr in CSV format

ftshwr -s -csv outputs a table with two rows indicating the number of requests that have the corresponding status.

Example

```
ftshwr –s –csv

Act;Wait;Lock;Susp;Hold;Fin;Total
0;1;0;0;2;0;3
```

10.4 BSFT commands

The BSFT interface is supported for the last time in the current version!

openFT for Unix systems also offers users the BSFT interface (Byte Stream File Transfer) for transferring and managing files. BSFT is a user interface for file transfer and file management standardized by X/Open. The definition of the interface is based on the CAE specification *Byte Stream File Transfer (BSFT)* from X/Open. For further information on the general definition of the BSFT interface, you should therefore refer to this specification.

The BSFT interface can be used to issue both file transfer and file management requests. The functions can be used to the extent that they are supported by the partner system.

10.4.1 BSFT session

BSFT is started with the command *bsft*. When started, it outputs the prompt *bsft>*. You are now in BSFT command mode and can issue file transfer requests and file management requests by means of special BSFT commands.

Before issuing the first file transfer or file management request, you must provide BSFT with information on the remote system and possibly the remote transfer admission.

You can specify the name of the desired remote system, either when calling BSFT, or later on in command mode using *open*.

When calling BSFT, you define whether the information on remote transfer admission is automatically retrieved by BSFT (auto-login) or to be specified by you in BSFT command mode.

By default, BSFT uses the auto-login function. Here, when a file transfer or file management request is issued, BSFT searches the *.ftamrc* file in the home directory for a transfer admission entry.

– If the file exists and a valid entry is found, the request is executed.

– If the file does not exist or is empty, BSFT queries the login name and the password after the request is issued. If you respond to the login name query by pressing the ↵ key, BSFT takes the login name under which it was called. If you respond to the password query by pressing the ↵ key, the request is executed without the password.

– If the file contains an entry without a password, BSFT queries the password. If you respond by pressing the ↵ key, the request is processed without a password.

You can deactivate the auto-login function when calling BSFT. In this case, you must use the *user* command to indicate the desired transfer admission, at the latest when issuing the first file transfer or file management request. Otherwise, a corresponding warning will be output. If transfer admission is required to execute a request, the request is rejected.

If you wish to communicate with a partner system that does not support login names, e.g. openFT(Windows), you must deactivate the "auto-login" function when calling BSFT.

If you wish to pass information automatically with auto-login, you must create the file *.ftamrc* in your home directory. Enter the desired transfer admission in the first line of the file.

The entry must be specified in the syntax of the remote system, see section “Transfer admission” on page 94.

The password can also be entered in hexadecimal form x‘...‘ or X‘...‘. This is of relevance for links to openFT for BS2000/OSD, because BS2000 supports the definition of hexadecimal passwords.

A connection to the remote system is not established until it is required by a file transfer or file management request.

You can cancel entries for the remote system using the *close* command.

You can exit BSFT command mode using the *bye* or *quit* command, or the END key.

10.4.2 Call BSFT

BSFT is started with the command *bsft*. When calling the program, you can select some BSFT functions by specifying the corresponding options, such as the name of the desired system, in the call. You then enter BSFT command mode, where you can issue further commands (BSFT commands).

Format

bsft[-**v**][-**i**][-**n**][-**g**][host]

Description

-v (Default value)
Activates display mode. In this case, the file type is also output with the confirmation of the *get* and *mget* commands (See BSFT command *verbose*).

-i Deactivates dialog mode.
-i not specified (default value)
Dialog mode is activated. When transferring, deleting, or displaying a number of files, therefore, an input request is issued after each file (see BSFT command *prompt*).

-n Deactivates auto-login. Using the *user* command, you must specify the transfer admission for the remote system before the first file transfer or file management request.

-n not specified (default value)
Activates auto-login. BSFT first searches the *.ftamrc* file for transfer admission. If the file does not exist or is empty, BSFT queries the login name and the password when the first request is issued. If you respond to the login name query by simply pressing the ↵ key, BSFT takes the login name under which it was called. If you respond to the password query by pressing the ↵ key, no password is supplied with the request (see section “BSFT session” on page 453).

-g Activates the expansion of file names (see BSFT command *glob*).

-g not specified (default value)
File names are not expanded.

host You may specify the host name in a variety of ways:
- as a TNS name (all 5 parts of the name are supported)
- as a TCP/IP host name;
  a port number can be specified after the host name. It must be separated from the host name by a colon (:*port*).
- as an IP address; with or without the prefix %ip:
  - with the prefix %ip (e.g. %ip139.22.33.44)
    In this case, the partner name is treated directly as an IP address. This approach is recommended when the hosts file is located on a different computer.
  - without prefix (e.g. 139.22.33.44)
    In this case, the TNS is searched first, followed by the file */etc/hosts*. If no matching entry is found there, the path name is treated as an IP address.

  A port number can be specified after the IP address just like it can be for the TCP/IP host name; it must be separated from the IP address by a colon (:*port*).

A distinction is made between uppercase and lowercase.

The remote system can also be specified after entering BSFT command mode (see BSFT command *open*).

When calling BSFT, you can automatically execute commands by placing them in a file, which is then read by BSFT.

Example

```
bsft -n <bsftscript
```

The contents of the file *bsftscript* may appear as follows:

```
open bs2r2
user suzanne secret 123456
get remotefile
bye
```

10.4.3 Functional overview of BSFT commands

This section is intended primarily to provide an overview of all BSFT commands, sorted according to their function.

Interrupt or terminate BSFT

bye, quit	Terminate BSFT session
!command	Execute command from the shell

Information on connection setup

open	Specify the name of the remote system
user	Specify the transfer admission for the remote system
password	Specify the transfer admission password for the remote system
account	Specify the transfer admission account number for the remote system
close	Cancel the specification of the remote system

File transfer

get, recv	Transfer a remote file to the local system
mget	Transfer a number of remote files to the local system
put, send	Transfer a local file to the remote system
mput	Transfer a number of local files to the remote system
append	Append a local file to a file in the remote system

Output information

dir	List the contents of a remote directory (long form)
mdir	List the contents of a number of remote directories (long form)
ls	List the contents of a remote directory
mls	List the contents of a number of remote directories
pwd	Display the current directory for the remote system
status	Display BSFT status
help, ?	Display information on BSFT commands

Working with files

cd	Change directory in the remote system
lcd	Change directory in the local system
delete	Delete a file in the remote system
mdelete	Delete a number of files in the remote system
rename	Rename file in the remote system

Job control

bell	Activate/deactivate acoustic signal
glob	Activate/deactivate file name expansion
prompt	Activate/deactivate dialog mode for multiple requests
verbose	Activate/deactivate display mode
encode	Define the default character set of file to be transferred
override	Define the default write mode in the receiving system
type	Define the default file type
binary	Set the default file type to binary
ascii	Set the default file type to text
create	Specify file password
separator	Define separator for directories

10.4.4 Notational conventions

The following conventions have been used for syntax diagrams:

Boldface
: stands for entries that must be written exactly as shown, e.g. command names.

Normal print
: stands for entries that must be replaced by the corresponding current values.

[ ]
: enclose optional entries. The effect on the function of the command is described for the individual parameters.

␣
: stands for at least one blank that must be inserted between the various entries. You may also enter more than one blank.

The commands and fixed values can be abbreviated to the extent that their short forms are still unique. However, these short forms are not guaranteed for subsequent versions, and should therefore not be used in procedures.

10.4.5 Command descriptions in alphabetical order

account - Specify account number for the remote system

This command is used to specify the account number for transfer admission. An account number is required if the remote system is not X/Open-compliant.

Format

account[␣account number]

Description

account number
: Account number.

account number not specified
: The previous entry for the account number is deleted.

append - Append a local file to a file in the remote system

This command appends a local file to a file in the remote system.

At file transfer, the current values for file type, character set, and write mode are used.

Format

append␣local file[␣remote file]

Description

local file
: Name of the local file to be appended.

remote file
: Name of the remote file to which the local file is to be appended.

remote file not specified
: The transferred data is appended to the *local file* in the remote system. If *local file* does not exist, it is created.

ascii - Set default file type to text

This command is used to set the default file type to *Text*. This means that the file is transferred as a text file.

It has the same effect as the command line *type ascii*.

Format

ascii

bell - Activate/deactivate acoustic signal

This command activates an acoustic signal which is output when a file transfer request is completed. The acoustic signal is deactivated when BSFT is started.

Format

bell[␣value]

Description

value
: Specifies whether the acoustic signal is activated or deactivated.

  on The acoustic signal is activated.

  off The acoustic signal is deactivated.

value not specified
: Switches to the alternative setting.

binary - Set default file type to binary

This command sets the default file type to *Binary*. This means that the file is transferred as a binary file.

It has the same effect as the command line *type binary*.

Format

binary

bye - Exit BSFT command mode

This command exits BSFT command mode. It has the same effect as the *quit* command and the END key.

Format

bye

cd - Change directory in the remote system

This command is used to change the current working directory in the remote system. The new directory comes into effect when the next file transfer or file management request is issued.

Format

cd ␣ remote directory

Description

remote directory
: Name of the directory to which you wish to change.

close - Cancel the specification of a remote system

This command is used to cancel the specification of the remote system. With the *open* command, you can then specify a new system to which a connection is to be established.

Format

close

create - Specify file password

This command is used to specify a file password for file transfer.

If the file in the remote system is protected by a password, this password must be specified with *create* when transferring a local file to the remote system.

Format

create[␣password]

Description

password
: File password.

  The password can also be entered in hexadecimal form x‘...‘ or X‘...‘. This is of relevance for links to openFT for BS2000/OSD, because BS2000 supports the definition of hexadecimal passwords.

password not specified
: BSFT requests you to enter the password. The local echo is deactivated beforehand. If you press the ↵ key without entering a password, the password previously specified is deleted.

delete - Delete file in the remote system

This command deletes the *remote file* on the remote system. You can only delete one file at a time with this command. If you wish to delete a number of files, use the *mdelete* command.

Format

delete␣remote file

Description

remote file
: Name of a file on the remote system.

dir - List the contents of a remote directory (long form)

This command returns detailed information on a remote directory. Output is normally written to the screen. If you specify a *local file* in the command, output is written to this file.

Format

dir[␣remote directory[␣local file]]

Description

remote directory
: Directory on the remote system whose contents are to be output.

remote directory not specified
: The contents of the current working directory on the remote system are output.

local file
: Name of the local file to which the contents of the directory are to be output.

local file not specified
: Output is written to the screen.

Example

```
bsft> dir

bsft: management request completed
tr-pxeacd       smith              203 Apr 28 16:46 .profile
dr     a        smith             8252 May 06 14:26 info
dr     a        smith             8252 May 30 09:15 test
```

The format corresponds to the standard output for the FT commands *ftshw* and *ftshwf*.

encode - Define the default character set for the data to be transferred

This command defines the character set for text files from which the data to be transferred is taken.

Format

encode␣direction[␣character set]

Description

direction

Indicates the direction of transfer for which the character set is to be defined. You can enter either “receive” or “transmit”.

receive

The character set applies to file transfer to the local system.

transmit

The character set applies to file transfer to the remote system.

character set

Specifies the character set for a text file. The following values are possible:

graphicstring	Default value for *transmit*
generalstring	
ia5	
visiblestring	
unknown	For *receive* only (default value for *receive*).

This entry is only relevant for FTAM partners.

character set not specified

The value *graphicstring* applies to file transfer to the remote system, and *unknown* applies to file transfer to the local system.

get - Transfer a remote file to the local system

This command fetches the *remote file* from the remote system, and stores it in the local system.
When transferring files, the current values for the file type, character set, and write mode are used, unless otherwise specified in the command line.

The *get* command has the same effect as the *recv* command.

Format

get␣remote file[␣local file][␣file type]

Description

remote file
: Specifies the name of the remote file to be retrieved.

local file
: Specifies the name to be assigned to the file in the local system.

local file not specified
: The local file name is the same as *remote file*.

file type
: Indicates the file type. For *file type*, you can enter either “ascii” or “binary”.

  -ascii[␣character set]
  : The *remote file* is fetched as a text file.

  In this case, you can define the character set of the text in the file.

  character set
  : Indicates the character set of the text in the file. Possible values are listed below:

  ia5, graphicstring, visiblestring, generalstring, unknown

  This entry is only relevant for FTAM partners.

  character set not specified
  : The current default character set applies.

  -binary
  : The *remote file* is fetched as a binary file.

file type not specified
: The current default file type applies.

glob - Activate/deactivate file name expansion

The command is used to activate/deactivate file name expansion.

If activated, each local file name or path name is processed on the basis of the shell special characters. For instance, with the command *mget a**, all files whose names begin with *a* are transferred. If file name expansion is deactivated, all file names and path names are handled as they are specified.

Format

glob[␣value]

Description

value
: Specifies whether file name expansion is to be activated or deactivated.

  on File name expansion is activated.

  off File name expansion is deactivated.

value not specified
: Switches to the alternative setting.

help - Display information on BSFT commands

This command is used to output the syntax of BSFT commands. It has the same effect as the *?* command.

Format

help[␣command]

Description

command
: Name of the command for which you require information.

command not specified
: A list of all BSFT commands is output.

lcd - Change directory in the local system

This command is used to change to another directory in the local system.

Format

lcd[␣directory]

Description

directory
: Name of the directory to which you wish to change.

directory not specified
: *lcd* changes to your home directory.

ls - List the contents of the remote directory

This command lists the contents of a directory in the remote system. Output is normally written to the screen. If you specify a *local file* in the command, however, output is written to this file.

The format corresponds to the default output for the FT commands *ftshw* and *ftshwf*.

Format

ls[␣remote directory[␣local file]]

Description

remote directory
: Name of the directory in the remote system.

remote directory not specified
: The current working directory is used.

local file
: Name of a local file to which output from *ls* is to be redirected.

local file not specified
: Output is written to the screen.

mdelete - Delete a number of files in the remote system

This command deletes a number of files on the remote system. *mdelete* asks you to confirm your request before deleting files. It then waits for confirmation or rejection of the request. This query can be suppressed using the *prompt* command.

Format

mdelete␣remote filei1[␣remote file2][...]

Description

remote file1[␣remote file2][...]
: Name of the files to be deleted. You can specify up to ten files at a time.

mdir - Output a number of remote directories to a file (long form)

This command is the same as the *dir* command, except that here you can specify up to ten directories at the same time. Output is written to the *local file*.

Format

mdir␣remote directory1[␣remote directory2][...]␣local file

Description

remote directory1, remote directory2, ...
: Name of a directory in the remote system. You can specify up to ten directories.

local file
: Name of a local file to which output is written.

mget - Transfer a number of files to the local system

This command fetches a number of files from the current directory on the remote system, and transfers them to the current local directory. Before the files are transferred, *mget* outputs a query and waits for you to confirm or reject each file.

This query can be suppressed using the *prompt* command.

If file name expansion is activated, the special characters are evaluated (see *glob* command).

Format

mget␣remote file1[␣remote file2][...]

Description

remote file1[␣remote file2][...]
: Names of the files to be fetched. You can specify up to ten files at a time.

mls - Output a number of remote directories to a file

This command is the same as the *ls* command. It can be used to specify up to ten directories at the same time. Output is written to the *local file*.

Format

mls␣remote directory1[␣remote directory2][...]␣local file

Description

remote directory1, remote directory2, ...
: Name of a directory in the remote system. You can specify up to ten directories.

local file
: Name of a local file to which output is written.

mput - Transfer a number of files to the remote system

This command transfers a number of local files to the current working directory of the remote system. Before *mput* transfers the files, you must confirm or reject each file. This query can be suppressed using the *prompt* command.

Format

mput␣local file1[␣local file2][...]

Description

local file1[␣local file2][...]

Names of the files to be transferred. Up to ten file names can be specified at the same time.

open - Specify the name of the remote system

This command is used to specify the remote system to which you wish to establish a connection. You can also specify the remote system name in the *host* parameter when starting BSFT.

Format

open␣host

Description

host

You may specify the name in a variety of ways:

- as a TNS name (all 5 parts of the name are supported)
- as a TCP/IP host name;
  a port number can be specified after the host name. It must be separated from the host name by a colon (:*port*).
- as an IP address; with or without the prefix %ip:
  - with the prefix %ip (e.g. %ip139.22.33.44)
    In this case, the partner name is treated directly as an IP address. This approach is recommended when the hosts file is located on a different computer.
  - without prefix (e.g. 139.22.33.44)
    In this case, the TNS is searched first, followed by the file */etc/hosts*. If no matching entry is found there, the partner name is treated as an IP address.

  A port number can be specified after the IP address just like it can be for the TCP/IP host name; it must be separated from the IP address by a colon (:*port*).

A distinction is made between uppercase and lowercase.

override - Define the default write mode in the receiving system

This command specifies whether the destination file is to be overwritten if it already exists. You can enter either “fail”, or “overwrite”.

Format

override␣value

Description

value
Possible values are listed below:

fail If the destination file already exists, the file transfer request is rejected.

overwrite (default value)
If the destination file already exists, the file is overwritten.

prompt - Activate/deactivate dialog mode for multiple requests

This command activates or deactivates dialog mode for multiple requests (*mdelete*, *mdir*, *mget*, *mls*, *mput*). If dialog mode is activated, BSFT waits for confirmation or rejection before executing each individual request. If dialog mode is deactivated, BSFT executes the requests without confirmation. For example, a multiple delete request (*mdelete*) will be executed without a query.

Format

prompt[␣value]

Description

value
: Specifies whether dialog mode is activated or deactivated.

  on Dialog mode is activated.

  off Dialog mode is deactivated.

value not specified
: Switches to the alternative setting.

By default, dialog mode is activated when you call BSFT. However, you can deactivate dialog mode when calling the BSFT program using the *-i* option.

put - Transfer a local file to the remote system

This command transfers a local file to the remote system. At file transfer, the current values for file type, character set, and write mode are used.

You can only transfer one file with *put*. If you wish to transfer a number of files, you should use the *mput* command.

The *put* command has the same effect as the *send* command.

Format

put␣local file[␣remote file][␣file type]

Description

local file
: Name of the file to be transferred.

remote file
: Name to be assigned to the file in the remote system.

remote file not specified
The remote file name is the same as *local file*.

file type
: Indicates the file type. For *file type*, you can enter either “ascii” or “binary”.

  -ascii[␣character set]

  The *local file* is sent as a text file. In this case, you can define the character set of the text in the file.

  character set

  : Indicates the character set of the text in the file. Possible values are listed below:
    ia5, graphicstring, visiblestring, generalstring
    This entry is only relevant for FTAM partners.

  character set not specified
  : The current default character set applies.

  -binary
  The *local file* is sent as a binary file.

file type not specified
: The current default file type applies.

pwd - Output current working directory for the remote system

This command outputs the name of the current working directory for the remote system.

Format

pwd

quit - Quit BSFT

This command terminates the BSFT session. It has the same effect as the *bye* command or the CTRL key.

Format

quit

recv - Transfer a remote file to the local system

This command fetches the *remote file* from the remote system and stores it in the local system.

It has the same effect as the *get* command.

You can only transfer one file with the *recv* command. If you wish to transfer a number of files, you should used the *mget* command.

At file transfer, the current values for file type, character set, and write mode are used, unless otherwise specified in the command line.

Format

recv␣remote file[␣local file][␣file type]

Description

remote file
: Name of the remote file to be fetched.

local file
: Name of the file on the local system.

local file not specified
: The local file name is the same as *remote file*.

file type
: Indicates the file type. For *file type*, you can enter either “ascii” or “binary”.

  -ascii[␣character set]
  The *remote file* is a text file. In this case, you can define the character set of the text in the file.

  character set

  : Indicates the character set of the text in the file. Possible values are listed below:
    ia5, graphicstring, visiblestring, generalstring, unknown
    This entry is only relevant for FTAM partners.

  character set not specified
  : The current default character set applies.

  -binary
  : The *remote file* is fetched as a binary file.

file type not specified
: The current default file type applies.

rename - Rename a file in the remote system

This command assigns a new name to a file on the remote system.

Format

rename␣remote file1␣remote file2

Description

remote file1
: Name of the file to be renamed.

remote file2
: New name to be assigned to the file.

send - Transfer a local file to the remote system

This command transfers a local file to the remote system. At file transfer, the current values for file type, character set, and write mode are used.

This command has the same effect as the *put* command.

You can only transfer one file with *send*. If you wish to transfer a number of files, you should use the *mput* command.

Format

send␣local file[␣remote file][␣file type]

Description

local file
: Name of the file to be transferred.

remote file
: Name of the file in the remote system.

remote file not specified
: The remote file name is the same as *local file*.

file type
: Indicates the file type. For *file type*, you can enter either “ascii” or “binary”.

  -ascii[␣character set]
  : The *local file* file is sent as a text file. In this case, you can define the character set of the text in the file.

  character set
  : Indicates the character set of the text in the file. Possible values are listed below:
    ia5, graphicstring, visiblestring, generalstring
    This entry is only relevant for FTAM partners.

  character set not specified
  : The current default character set applies.

  -binary
  : The *local file* is sent as a binary file.

file type not specified
: The current default file type applies.

separator - Define a separator for directories

This command is used to define a separator for directories. This definition is then taken into account when building a path name for the remote system. The defined separator is automatically appended to the directory specified with the *cd* command for the remote system, before the name of the file to be transferred is added.

Format

separator ␣character

Description

character
: Specifies the separator. You can only specify one character. The period character (.) is not permitted as a separator.

character not specified
: The slash (/) is used as the separator.

status - Display BSFT status

This command displays the current status of BSFT. This command can be used to view the current settings. Only fields that have been assigned values are output.

Format

status

Example

```
bsft> status

user-name: smith
local-directory: /home/williams
remote-directory: .
override: overwrite   separator: /
autologin: on   bell: off  glob: off  prompt: on   verbose: on
type: ascii  encoding from: unkown   encoding to: graphicstring
```

type - Define standard file type

This command is used to define a default file type for transferring files. You can enter either “ascii” or “binary”.

Format

type␣value

Description

value

The following values can be specified:

ascii (default value)

The file is a text file.

binary

The file is a binary file.

The command line *type ascii* has the same effect as the *ascii* command, and *type binary* has the same effect as the *binary* command.

user - Specify the transfer admission for the remote system

This command is used to specify a login name for the remote system.

It can also be used to delete previous entries by specifying this command without parameters and responding to the login name and password queries with the ↵ key. This may be useful, for example, if you wish to communicate with a partner system that does not support login names (e.g. openFT(Windows)).

If auto-login is switched on, this command has no effect.

Format

user[␣userid[␣password[␣account number]]]

Description

userid
: Login name of the user with whom you wish to communicate.

password
: Password of the user. If this is expected but not entered, BSFT will specifically ask you to input the password. Local echo is switched off beforehand. If you simply press the ↵ key without further entries, no password is specified.
The password can also be entered in hexadecimal form x'...' or X'...'. This is of relevance for links to openFT for BS2000, because BS2000 supports the definition of hexadecimal passwords.
Please note that the single quotes in BS2000 passwords need not be escaped with a backslash (\) here.

account number
: An account number is only required if you wish to communicate with systems that use account numbers. This is not the case with Unix systems.

verbose - Activate/deactivate display mode

This command activates or deactivates display mode. If the function is activated, the file type is output with the confirmation of the *get* and *mget* commands.

You can activate display mode when calling BSFT using the *-v* option (default value).

Format

verbose[␣value]

Description

value
: The following values can be specified:

  on Display mode is activated.

  off Display mode is deactivated.

value not specified
: Switches to the alternative setting.

? - Display information on BSFT commands

With this command, the syntax of BSFT commands is output to the screen.

Format

?[command]

Description

command
: Name of the command for which information is required.

command not specified
: A list of all BSFT commands is output.

The *?* command has the same effect as the *help* command.

! - Execute shell command

command is executed as a shell command in the local system.

Format

!command

Description

command
: Command to be executed as a shell command.

10.4.6 Messages from BSFT

This section lists the messages that you can receive in BSFT command mode. *$s* is replaced by the current value in the messages.

01 bsft: syntax error: the value > $s < is too much

02 bsft: syntax error: the value > $s < is invalid

03 bsft: syntax error: the parameter > $s < is mandatory

04 bsft: host already exists

05 bsft: host is removed

06 bsft: there isn‘ t any host

07 bsft: syntax error: the command > $s < is undefined

08 bsft: > $s < is the bsft-responder

09 bsft: > $s < is the bsft-user-account

10 bsft: > $s < is the bsft-user-name

11 bsft: > $s < is the bsft-user-password

12 bsft: syntax error: > $s < is an invalid -asciil-binary-value

13 bsft: syntax error: > $s < is an invalid on|off-value

14 bsft: syntax error: > $s < is an invalid contents-type

15 bsft: syntax error: > $s < is an invalid receive|transmit-value

16 bsft: syntax error: no encoding when > -binary < is set

17 bsft: syntax error: after encoding no more parameter

18 bsft: syntax error: invalid parameter-structure in trace-command

19 bsft: system error: internal-command > $s < too long

20 bsft: system error: insert-code > $s <

21 bsft: syntax error: the parameter > $s < is too long

22 bsft: receiving file > $s < already exists

23 bsft: syntax error: > $s < is an invalid fail|overwrite-value

24 bsft: $s $s

25 bsft: input-string > $s < is too long

26 bsft: syntax error: too many parameters

27 bsft: put in > $s < please

28 bsft: > $s < command doesn‘ t exist

29 bsft> $s

30 bsft: > $s < command is not implemented

31 bsft: > $s < option is not implemented

32 bsft: > $s < is missing; use the > $s < command first

33 bsft: > $s < syntax error in the file-content of > .ftamrc <

34 bsft: the file > $s < doesn‘ t exist

35 bsft: syntax error: the option > $s < is invalid

36 bsft: local directory is now: $s

37 bsft: remote directory is now: $s

38 bsft: not a directory: $s

39 host: $s

40 user-name: $s

41 user-account: $s

42 local-directory: $s

43 remote-directory: $s

44 account-name: $s

45 password: $s

46 user-name: $s

47 bsft: file < $s > doesn‘ t exist

48 bsft: lower layer system error

49 bsft: no such internal-message is used from bsft

50 bsft: internal syntax-error

51 bsft: get command completed

52 bsft: put command completed

53 bsft: illegal parameter value in internal command

54 bsft: interrupted from the remote system

55 bsft: status of local file inconsistent

56 bsft: local file gets no space

57 bsft: local file not accessible

58 bsft: local file resulting pathname is too long

59 bsft: local file I/O error

60 bsft: remote transfer/management admission invalid

61 bsft: local operating system error

62 bsft: remote resource shortage

63 bsft: remote system error

64 bsft: connection shortage

65 bsft: local system unknown in the remote system

66 bsft: connection lost

67 bsft: host is an unknown responder system

68 bsft: no data traffic within a certain time

69 bsft: local system error

70 bsft: error on remote file system

71 bsft: status of remote file inconsistent

72 bsft: remote file gets no space

73 bsft: remote file not accessible

74 bsft: too long path name for remote file

75 bsft: remote file I/O error

76 bsft: syntax error in the remote file name

77 bsft: internal error

78 bsft: protocol error

79 bsft: Restart not supported

80 bsft: OSS error

81 bsft: CMX error

82 bsft: internal administration error

83 bsft: internal trace function switched on

84 bsft: internal trace function switched off

85 bsft: file could not be deleted

86 bsft: attributes could not be modified

87 bsft: directory not found

88 bsft: directory not accessible

89 bsft: information output canceled

90 bsft: connection rejected or disconnected

91 bsft: wildcard expansion for directories not implemented

92 bsft: management call successful

93 bsft: outfile > $s < cannot be opened

94 bsft: syntax error: the command > $s < is ambiguous

95 bsft: syntax error: the parameter > $s < is ambiguous

96 bsft: is started with > $s <

97 bsft: remote system not accessible

98 bsft: disconnected from remote system

99 bsft: local resource shortage

100 bsft: management-request completed

101 bsft: future filesize not supported

102 bsft: legal qualification not supported

103 bsft: account not supported

104 bsft: file availability not supported

105 bsft: access mode not supported

106 bsft: two parameters are not allowed at the same time

107 bsft: command completed

108 bsft: warning: no .ftamrc because no home-directory

109 bsft: warning: no .ftamrc because name of home-directory is too long

110 bsft: remote file > $s < not found

111 bsft: remote file > $s < already exists

112 bsft: remote files > $s < not found

113 bsft: local file > $s < not found

114 bsft: local files > $s < not found

115 bsft: remote directories > $s < not found

116 bsft: transfer-request with file-type > $s < completed

117 bsft: > $s < permission denied by local system

118 bsft: > $s < function not supported in remote system

119 bsft: command without effect because autologin

120 bsft: separator not changed because in use

121 bsft: > $s < is an invalid separator

122 bsft: > $s < without any parameter not supported

123 bsft: > username< may be missing; case use the > user< command first

124 bsft: command without effect because account-name is in .ftamrc

125 bsft: creation of local file > $s < not authorized

126 bsft: local file > $s < FILE locked from concurrent access

127 bsft: command for file type not supported

128 bsft: character set of source file not supported by destination system

129 bsft: creation of remote file > $s < not authorized

130 bsft: remote file > $s < locked from concurrent access

131 bsft: invalid file password for remote file > $s <

132 bsft: remote file > $s < not shareable

133 bsft: illegal access mode to remote file > $s <

134 bsft: file owner unknown for remote file > $s <

135 bsft: retention period not yet expired for remote file > $s <

136 bsft: file transfer completion error

137 bsft: recovery failed. Code > $s <

138 bsft: No information available

139 bsft: function not supported in remote system or protocol error

140 bsft: function not supported by local FTAM

141 bsft: Management request completed

142 bsft: FTAM not started

143 bsft: FTAM not active

144 bsft: No space left on device

145 bsft: No authorization for this action

146 bsft: openFT not started

Glossary

Italic type indicates a reference to other terms in this glossary.

absolute path name

The entire path name, from the root directory to the file itself.

access control

File attribute in the *virtual filestore*, attribute of the *security group* that defines *access rights*.

access protection

Comprises all the methods used to protect a data processing system against unauthorized system access.

access right

Derived from the *transfer admission*. The access right defines the scope of access for the user who specifies the transfer admission.

action list

Component of the file attribute *access control* (attribute of the *security group*) in the *virtual filestore* that defines *access rights*.

ADM administrator

Administrator of the *remote administration server*. This is the only person permitted to modify the configuration data of the remote administration server.

ADM partner

Partner system of an openFT instance with which communication takes place over the *FTADM protocol* in order to perform *remote administration*.

ADM traps

Short messages sent to the *ADM trap server* if certain events occur during operation of openFT.

ADM trap server

Server that receives and permanently stores the *ADM traps*. It must be configured as a *remote administration server*.

administrated openFT instance

openFT instances that are able to be administered by *remote administrators* during live operation.

admission profile

Way of defining the *FTAC* protection functions. Admission profiles define a *transfer admission* that has to be specified in *FT requests* instead of the *LOGON or Login authorization*. The admission profile defines the *access rights* for a user ID by restricting the use of parameters in *FT requests*.

admission profile, privileged

-> see *privileged admission profile*

admission set

In *FTAC*, the admission set for a particular user ID defines which FT functions the user ID may use and for which *partner systems*.

admission set, privileged

-> see *privileged admission set*

AES (Advanced Encryption Standard)

The current symmetrical encryption standard, established by NIST (National Institute of Standards and Technology), based on the Rijndael algorithm, developed at the University of Leuven (B). The openFT product family uses the AES method to encrypt the request description data and possibly also the file contents.

ANSI code

Standardized 8-bit character code for message exchange. The acronym stands for "American National Standards Institute".

API (Application Program Interface)

An interface that is freely available to application programmers. It provides a set of interface mechanisms designed to support specific functionalities.

Application Entity Title (AET)

The Application Entity Title consists of Layer 7 addressing information of the *OSI Reference Model*. It is only significant for *FTAM partners*.

asynchronous request

Once the *FT request* has been submitted, it is processed independently of the user. The user can continue working once the system has confirmed acceptance of the request. (see also *synchronous request*).

authentication

Process used by openFT to check the unique identity of the request partner.

background process

A process that runs independently of the user process. A background process is started by placing the special character & at the end of a command. The process which initiates the background process is then immediately free for further tasks and is no longer concerned with the background process, which runs simultaneously.

basic functions

Most important file transfer functions. Several basic functions are defined in the *admission set* which can be used by a login name. The six basic functions are:

– inbound receive
– inbound send
– inbound follow-up processing
– inbound file management
– outbound receive
– outbound send

central administration

Central administration in openFT incorporates the *remote administration* and *ADM traps* functions and requires the use of a *remote administration server*.

character repertoire

Character set of a file in the *virtual filestore*.

In the case of files transferred with *FTAM partners* it is possible to choose between: *GeneralString*, *GraphicString*, *IA5String* and *VisibleString*.

client

– Term derived from client/server architectures: the partner that makes use of the services provided by a *server*.
– Logical instance which submits requests to a *server*.

cluster

A number of computers connected over a fast network and which in many cases can be seen as a single computer externally. The objective of clustering is generally to increase the computing capacity or availability in comparison with a single computer.

Comma Separated Value (CSV)

This is a quasi-tabular output format that is very widely used in the PC environment in which the individual fields are separated by a semicolon ";". It permits the further processing of the output from the most important openFT commands using separate tools.

communication controller

-> see *preprocessor*

compression

This means that several identical successive characters can be reduced to one character and the number of characters is added to this. This reduces transfer times.

computer network, open

-> see *open computer network*

concurrency control

Component of the FTAM file attribute *access control* (part of the *security group*) in the *virtual filestore* that controls concurrent access.

connectivity

In general, the ability of systems and partners to communicate with one another. Sometimes refers simply to the communication possibilities between transport systems.

constraint set

Component of the *document type*.

contents type

File attribute in the *virtual filestore,* attribute of the *kernel group* that describes the file structure and the form of the file contents.

data communication system

Sum of the hardware and software mechanisms which allow two or more communication partners to exchange data while adhering to specific rules.

data compression

Reducing the amount of data by means of compressed representation.

data encoding

Way in which an *FT system* represents characters internally.

Data Encryption Standard (DES)

International data encryption standard for improved security. The DES procedure is used in the FT products to encrypt the request description data and possibly the request data if connections are established to older versions of openFT that do not support *AES*.

data protection

- In the narrow sense as laid down by law, the task of protecting personal data against misuse during processing in order to prevent the disclosure or misappropriation of personal information.
- In the wider sense, the task of protecting data throughout the various stages of processing in order to prevent the disclosure or misappropriation of information relating to oneself or third parties.

data security

Technical and organizational task responsible for guaranteeing the security of data stores and data processing sequences, intended in particular to ensure that

- only authorized personnel can access the data,
- no undesired or unauthorized processing of the data is performed,
- the data is not tampered with during processing,
- the data is reproducible.

DHCP

Service in TCP/IP networks that automatically assigns IP addresses and TCP/IP parameters to clients on request.

directory

Directories are folders in the hierarchical file system of a Unix system (including POSIX) or a Windows system that can contain files and/or further directories.

document type

Value of the file attribute *contents type* (attribute of the *kernel group*). Describes the type of file contents in the *virtual filestore*.
– *document type* for text files: FTAM-1
– *document type* for binary files: FTAM-3

EBCDIC

Standardized code for message exchange as used in BS2000/OSD. The acronym stands for "Extended Binary Coded Decimal Interchange Code".

emulation

Components that mimic the properties of another device.

entity

-> see *instance*

Explorer

A program from Microsoft that is supplied with Windows operating systems to facilitate navigation within the file system.

file attributes

A file's properties, for example the size of the file, access rights to the file or the file's record structure.

file management

Possibility of managing files in the remote system. The following actions are possible:
– Create directories
– Display and modify directories
– Delete directories
– Display and modify file attributes
– Rename files
– Delete files.

filestore, virtual

-> see *virtual filestore*

file transfer request

-> see *FT- request*

firewall processor

Processor which connects two networks. The possible access can be controlled precisely and also logged.

fixed-length record

A record in a file all of whose records possess the same, agreed length. It is not necessary to indicate this length within the file.

follow-up processing

FT function that initiates execution of user-specified commands or statements in the *local* and/or the *remote system* after an *FT request* has been completed. The user may define different follow-up processing, depending on the success or failure of FT request processing. See also *preprocessing* and *postprocessing*.

follow-up processing request

Statements contained within an *FT request* for *follow-up processing* to be performed after file transfer.

FT administrator

Person who administers the openFT product installed on a computer. openFT can be administered from all login names with UID=0.

FT request

Request to an *FT system* to transfer a file from a *sending system* to a *receive system* and (optionally) start *follow-up processing requests*.

FT system

System for transferring files that consists of a computer and the software required for file transfer.

FT trace

Diagnostic function that logs FT operation.

FTAC (File Transfer Access Control)

Extended access control for file transfer and file management. In the case of BS2000 and z/OS, this is implemented by means of the product openFT-AC, for other operating systems it is a component of the openFT product, e.g. in openFT for Unix systems or openFT for Windows systems.

FTAC administrator

Administrator of the FTAC functions; should be identical to the person responsible for data security in the system.

FTAC logging function

Function which FTAC uses to log each access to the protected system via file transfer.

FTADM protocol

Protocol used for communication between two openFT instances in order to perform *remote administration* or transfer *ADM traps*.

FTAM-1

document type for text files

FTAM-3

document type for binary files

FTAM catalog

The FTAM catalog is used to extend the file attributes available in Unix systems. It is only relevant for access using FTAM. For example, a file can be deleted using the command *rm* on a Unix system, even if the *permitted actions* parameter does not allow this.

FTAM file attributes

All systems which permit file transfer via FTAM protocols must make their files available to their partners using a standardized description (ISO 8571). To this end, the attributes of a file are mapped from the physical filestore to a *virtual filestore* and vice versa. This process distinguishes between three groups of file attributes:

– kernel group: describes the most important file attributes.
– storage group: contains the file's storage attributes.
– security group: defines security attributes for file and system access control.

FTAM partner

Partner system that uses *FTAM protocols* for communication.

FTAM protocol (File Transfer, Access and Management)

Protocol for file transfer standardized by the “International Organization for Standardization” (ISO) (ISO 8571, FTAM).

FTP partner

Partner system that uses *FTAM protocols* for communication.

FTP protocol

Manufacturer-independent protocol for file transfer in TCP/IP networks.

functional standard

Recommendation defining the conditions and the forms of application for specific ISO standards (equivalent term: *profile*). The transfer of unstructured files is defined in the European Prestandard CEN/CENELEC ENV 41 204; file management is defined in the European Prestandard CEN/ CENELEC ENV 41205.

gateway

Generally understood to mean a computer that connects two or more networks and which does not function as a bridge. Variants: gateway at network level (= router or OSI relay), transport and application gateway.

gateway processor

Communication computer that links a computer network to another computer network. The mapping of the different protocols of the various computer networks takes place in gateway processors.

general string

Character repertoire for file files transferred to and from *FTAM partners*.

GraphicString

Character repertoire for files transferred to and from *FTAM partners*.

heterogeneous network

A network consisting of multiple subnetworks functioning on the basis of different technical principles.

homogeneous network

A network constructed on the basis of a single technical principle.

HOSTS file

Network administration file that contains the Internet addresses, the processor names and the alias names of all accessible computers.

IA5String

Character repertoire for files transferred to and from *FTAM partners*.

identification

Procedure making it possible to identify a person or object.

inbound file management

Request issued in a *remote system* for which directories or file attributes of the *local system* can be displayed, file attribute modified or local file deleted.

inbound follow-up processing

Request issued in a *remote system* with *follow-up processing* in the *local system*.

inbound receive

Request issued in the *remote system*, for which a file is received in the *local system*.

inbound request / inbound submission

Request issued in another system, i.e. for this request.

inbound send

Request issued in a *remote system* for which a file is sent from the *local system* to the remote system.

initiator

Here: *FT system* that submits an *FT request*.

instance / entity

A concept of OSI architecture: active element in a layer. Also see *openFT instance*.

instance ID

A network-wide, unique address of an openFT instance.

integrity

Unfalsified, correct data following the processing, transfer and storage phases.

interoperability

Capability of two *FT systems* to work together.

ISO/OSI reference model

The ISO/OSI Reference Model is a framework for the standardization of communications between open systems. (ISO=International Standards Organization).

job

Sequence of commands, statements and data.

job transfer

Transfer of a file that constitutes a *job* in the *receive system* and is initiated as a job there.

kernel group

Group of file attributes of the *virtual filestore* that encompasses the kernel attributes of a file.

library

File with internal structure (elements)

library element

Part of a library. A library element may in turn be subdivided into a number of records.

Local Area Network (LAN)

Originally a high-speed network with limited physical extension. Nowadays, any network, that uses CSMA/CD, Token Ring or FDDI irrespective of the range (see also *WAN Wide Area Network*).

local system

The *FT system* at which the user is working.

logging function

Function used by openFT to log all file transfer accesses to the protected system.

log record

Contains information about access checks performed by openFT (FTAC log record) or about a file transfer or remote administration request which is started when the access check was successful (FT log record or ADM log record).

Logical Unit (LU)

Interface between an application program and the SNA data communications network. The LU type describes the communications characteristics.

Login authorization

Transfer admission to a computer which (as a rule) consists of the login name and the password, and authorizes dialog operation, see also *LOGON authorization*.

LOGON authorization

Transfer admission authorizing access to a computer. The LOGON authorization (normally) consists of user ID, account number and password and authorizes the user to make use of interactive operation.

mailbox

The mailbox is a file which is read using the mail command. Each user has a mailbox for receiving messages.

maximum-string-length

Specifies the maximum length of *strings* within a file in the *virtual FTAM filestore*.

Network Control Program (NCP)

Operating system of the front-end-processor for SNA hosts.

network description file

File used up to openFT V9 that contains specifications concerning *remote systems* (*FT systems*).

open computer network

Computer network in which communication is governed by the rules of ISO/OSI. Interoperation of different computers from various vendors is made possible by defined *protocols*.

openFT Explorer

openFT program that provides a graphical user interface that allows file transfer and administration functions to be performed.

openFT instance

Several openFT systems, so-called openFT instances, can be running simultaneously on a cluster of a TCP/IP network. Each instance has its own address (instance ID) and is comprised of the loaded code of the openFT products (including add-on products if they are available) and of the variable files such as partner list, logging files, request queue, etc.

openFT Monitor

Program that allows the monitoring data for openFT operation to be shown in the form of a chart. openFT Monitor requires a graphics-capable terminal.

openFT partner

Partner system which is communicated with using *openFT protocols*.

openFT protocols

Standardized *protocols* for file transfer (SN77309, SN77312).

openFT-FTAM

Add-on product for openFT (for BS2000, Unix systems and Windows systems) that supports file transfer using FTAM protocols. FTAM stands for File Transfer, Access and Management (ISO 8571).

openFT-Script

openFT interface providing an XML based script language that includes file transfer and file management functions. This interface allows you to combine several file transfer or file management requests to form a single openFT-Script request.

openFT-Script commands

Commands used for administering openFT-Script requests.

operating parameters

Parameters that control the *resources* (e.g. the permissible number of connections).

outbound request / outbound submission

Request issued in your own processor.

outbound receive

Request issued locally for which a file is received in the *local system*.

outbound send

Request issued locally for which a file is sent from the *local system*.

owner of an FT request

Login name in the *local system* or *remote system* under which this *FT request* is executed. The owner is always the ID under which the request is submitted, not the ID under which it is executed.

partner

-> see *partner system*

partner list

File containing specifications concerning *remote systems* (*FT systems*).

partner system

Here: *FT system* that carries out *FT requests* in cooperation with the *local system*.

password

Sequence of characters that a user must enter in order to access a user ID, file, job variable, network node or application. The user ID password serves for user *authentication*. It is used for access control. The file password is used to check access rights when users access a file (or job variable). It is used for file protection purposes.

permitted actions

File attribute in the *virtual filestore*; attribute of the *kernel group* that defines actions that are permitted in principle.

port number

Number that uniquely identifies a TCP/IP application or the end point of a TCP/IP connection within a processor.

POSIX (Portable Open System Interface)

Board and standards laid down by it for interfaces that can be ported to different system platforms.

postprocessing

openFT makes it possible to process the received data in the receiving system through a series of operating system commands, under the process control of openFT (in contrast to *follow-up processing*).

preprocessing

The preprocessing facility in openFT can be used to send a receive request in which the outputs of a remote command or program are transferred instead of a file. This makes it possible to query a database on a remote system, for example. Preprocessing also may be issued locally.

presentation

Entity that implements the presentation layer (layer 6) of the *ISO/OSI Reference Model* in an *FT system* that uses e.g. *FTAM protocols*.

presentation selector

Subaddress used to address a *presentation application*.

private key

Secret decryption key used by the recipient to decrypt a message that was encrypted using a *public key*. Used by a variety of encryption procedures including the *RSA procedure*.

privileged admission profile

Admission profile that allows the user to exceed the *FTAC administrator's* presettings in the *admission set*. This must be approved by the *FTAC administrator* who is the only person able to privilege admission profiles.

privileged admission set

Admission set belonging to the *FTAC administrator*.

profile

In OSI, a profile is a standard which defines which protocols may be used for any given purpose and specifies the required values of parameters and options.

Here: a set of commands assigned to a user ID. The permissibility of these commands is ensured by means of syntax files. See also *admission profile*, *privileged admission profile*.

prompting in procedures

Function used to prompt the user at the terminal to enter data required to run the procedure.

protocol

Set of rules governing information exchange between peer partners in order to achieve a defined objective. This usually consists of a definition of the messages that are to be exchanged and the correct sequencing of messages including the handling of errors and other exceptions.

public key

Public encryption key defined by the receiver of a message, and made public or made known to the sender of the message. This allows the sender to encrypt messages to be sent to the receiver. Public keys are used by various encryption methods, including the *Rivest Shamir Adleman (RSA) procedure*. The public key must match the *private key* known only to the receiver.

RAS

Remote Access Service; a Windows service that enables communication with remote systems.

receive file

File in the *receive system* in which the data from the *send file* is stored.

receive system

System to which a file is sent. This may be the *local system* or the *remote system*.

record

Set of data that is treated as a single logical unit.

relative path name

The path from the current *directory* to the file.

remote administration

Administration of openFT instances from remote computers.

remote administration server

Central component required for *remote administration* and for *ADM traps*. A remote administration server runs on a Unix or Windows system running openFT as of V11.0. If it is used for *remote administration*, it contains all the configuration data required for this purpose.

remote administrator

Role configured on the *remote administration server* and which grants permission to execute certain administration functions on certain openFT instances.

remote system

-> see *partner system*

request

Here: *FT request*

request queue

File containing *asynchronous requests* and their processing statuses.

request identification / request ID

number that identifies an *FT request.*

request management

FT function responsible for managing *FT requests*; it ensures request processing from the submission of a request until its complete processing or termination.

request number

-> see *request identification*

request storage

FT function responsible for storing *FT requests* until they have been fully processed or terminated.

resources

Hardware and software components needed by the *FT system* to execute an *FT request* (processes, connections, lines). These resources are controlled by the *operating parameters*.

responder

Here: *FT system* addressed by the *initiator*.

restart

Automatic continuation of an *FT request* following an interruption.

restart point

Point up to which the data of the *send file* has been stored in the *receive file* when a file transfer is interrupted and at which the transfer of data is resumed following a *restart*.

result list

List with information on a completed file transfer. This is supplied to the user in the *local system* and contains information on his or her *FT requests*.

RFC (Request for Comments)

Procedure used on the Internet for commenting on proposed standards, definitions or reports. Also used to designate a document approved in this way.

RFC1006

Supplementary protocol for the implementation of ISO transport services (transport class 0) using TCP/IP.

Rivest-Shamir-Adleman-procedure (RSA procedure)

Encryption procedure named after its inventors that operates with a key pair consisting of a *public key* and a *private key*. Used by the openFT product family in order to reliably check the identity of the partner system and to transmit the AES key to the partner system for encrypting the file contents.

router

Network element that is located between networks and guides message flows through the networks while simultaneously performing route selection, addressing and other functions. Operates on layer 3 of the OSI model.

security attributes

An object's security attributes specify how and in what ways the object may be accessed.

Secure FTP

Method by which a connection is tunneled using the *FTP protocol*, thus allowing secure connections with encryption and *authentication*.

security group

Group of file attributes in the *virtual filestore*, encompassing the security attributes of a file.

security level

When *FTAC functions* are used, the security level indicates the required level of protection against a *partner system*.

send file

File in the *sending system* from which data is transferred to the *receive file*.

sending system

Here: *FT system* that sends a file. This may be the *local system* or the *remote system*.

server

Logical entity or application component which executes a client's requests and assures the (coordinated) usage of all the generally available services (File, Print, data base, Communication, etc.). May itself be the client of another server.

service

- As used in the OSI architecture: a service is the set of functions that a service provider makes available at a service access point.
- As used in the client/server architecture: a set of functions that a server makes available to its clients.
- Term used in Windows: A program, routine or process used to perform a particular system function to support other programs, in particular on a low level (hardware-related).

service class

Parameter used by *FTAM partners* to negotiate the functions to be used.

session

- In OSI, the term used for a layer 5 connection.
- In SNA, a general term for a connection between communication partners (applications, devices or users).

session selector

Subaddress used to address a *session* application.

shell metacharacters

The following metacharacters have special meanings for the shell: *, [ ], ?, <, >, |, &, &&, ( ), { }

SNA network

Data communication system that implements the Systems Network Architecture (SNA) of IBM.

SNMP (Simple Network Management Protocol)

Protocol for TCP/IP networks defined by the Internet Engineering Task Force (IETF) for the transfer of management information.

special characters

-> see *shell metacharacters.*

standard admission set

This standard admission set applies by default to all users for whom there is no dedicated admission set. These default settings may be restricted further by the user for his or her own admission set.

standard error output (stderr)

By default, standard error output is to the screen.

standard input (stdin)

By default, standard input is from the keyboard.

standard output (stdout)

By default, standard output is to the screen.

storage group

File attribute in the *virtual filestore*, encompasses the storage attributes of a file.

string

Character string

string significance

Describes the format of *strings* in files to be transferred using *FTAM protocols*.

synchronous request

The user process that submitted the *FT request* waits for transfer to terminate. The user cannot continue working (see also *asynchronous request*).

system

-> see *FT- system*

system, local

-> see *local system*

system, remote

-> see *remote system*

TCP/IP (Transmission Control Protocol / Internet Protocol)

Widely used data transmission protocol (corresponds approximately to layers 3 and 4 of the *ISO/OSI reference model*, i.e. network and transport layers); originally developed for the ARPANET (computer network of the US Ministry of Defense) it has now become a de-facto standard.

transfer admission

Authorization for file transfer and file management when using FTAC. The transfer admissions is then used in place of the *LOGON or LOGIN authorization*.

transfer identification

-> see *request identification*

transfer unit

In an FTAM environment, the smallest data unit for transporting file contents. For *FTAM-1* and *FTAM-3* these are *strings*. A transfer unit can, but need not, correspond to one file record.

Transmission Control Protocol / Internet Protocol

-> see *TCP/IP*

TranSON

TranSON is a software product that permits secure access to a server. The use of TranSON is transparent to the application. The connection to the remote partner goes from the workstation through a client proxy and server proxy to the remote partner. The client proxy is located on the workstation, and the server proxy is located on the remote partner. The data transferred between the client proxy and the server proxy is encrypted.

transport connection

Logical connection between two users of the transport system (terminals or applications).

transport layer

Layer 4 of the *ISO/OSI reference model* on which the data transport protocols are handled.

Transport Name Service (TNS)

Service used to administer properties specific to transport systems. Entries for *partner systems* receive the information on the particular *transport system* employed.

transport protocol

Protocol used on the *transport layer*

transport selector (T-selector)

Subaddress used to address an ISO-8072 application in the *transport layer*.

transport system

- The part of a system or architecture that performs approximately the functions of the four lower OSI layers, i.e. the transport of messages between the two partners in a communication connection.
- Sum of the hardware and software mechanisms that allow data to be transported in computer networks.

Unicode

The universal character encoding, maintained by the Unicode Consortium. This encoding standard provides the basis for processing, storage and interchange of text data in any language in all modern software and information technology protocols. The Unicode Standard defines three Unicode encoding forms:
UTF-8, UTF-16 and UTF-32.

universal-class-number

Character repertoire of a file in the *virtual filestore*.

UNIX®

Registered trademark of the Open Group for a widespread multiuser operating system. A system may only bear the name UNIX if it has been certified by the Open Group.

Unix system

Commonly used designation for an operating system that implements functions typical of UNIX® and provides corresponding interfaces. POSIX and Linux are also regarded as Unix systems.

variable length record

A record in a file all of whose records may be of different lengths. The record length must either be specified in a record length field at the start of the record or must be implicitly distinguishable from the next record through the use of a separator (e.g. Carriage Return - Line Feed).

virtual filestore

The FTAM virtual filestore is used by *FT systems* acting as *responders* to make their files available to their *partner systems*. The way a file is represented in the virtual filestore is defined in the FTAM standard, see *file attributes*.

visibleString

Character repertoire for files transferred to and from *FTAM partners*.

WAN (Wide Area Network)

A public or private network that can span large distances but which runs relatively slowly and with higher error rates when compared to a *LAN*. Nowadays, these definitions have only limited validity. Example: in ATM networks.

X terminal

A terminal or software component to display the graphical X Window interface of Unix systems. An X terminal or a corresponding software emulation is a prerequisite for using the graphical interface of openFT.

Abbreviations

ACSE	Association Control Service Element
AES	Advanced Encryption Standard
AET	Application Entity Title
ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
BCAM	Basic Communication Access Method
BSFT	Byte Stream File Transfer
CAE	Common Application Environment
CEN	Comité Éuropéen de Normalisation
CENELEC	Comité Éuropéen de Normalisation Électrotechnique
CMX	Communication Manager Unix Systems
CCP	Communication Control Programm
DCAM	Data Communication Access Method
DCM	Data Communication Method
DES	Data Encryption Standard
DIN	Deutsches Institut für Normung (German standards institute)
DNS	Domain Name Service
EBCDIC	Extended Binary-Coded Decimal Interchange Code
ENV	Europäischer Normen-Vorschlag (European prestandard)
FADU	File Access Data Unit

FJAM	File Job Access Method
FSB	Forwarding Support Information Base
FSS	Forwarding Support Service
FT	File Transfer
FTAC	File Transfer Access Control
FTAM	File Transfer, Access and Management (ISO 8571)
FTPS	FTP via SSL / TLS
GPL	Gnu Public License
GSM	Global System for Mobile Communication
ISAM	Index Sequential Access Method
ISO	International Organization for Standardization
LAN	Local Area Network
LMS	Library Maintenance System
MSV	Mittelschnelles Synchron Verfahren (Medium-fast synchronous method)
NDMS	Network Data Management System
NIS	Network Information Service
OSI	Open Systems Interconnection
OSS	OSI Session Service
PAM	Pluggable Authentication Modules
PICS	Protocol Implementation Conformance Statement
PLAM	Primary Library Access Method

RFC1006	Request for Comments 1006
SAM	Sequential Access Method
SDF	System Dialog Facility
SNA	Systems Network Architecture
SNMP	Simple Network Management Protocol
SNPA	Subnetwork Point of Attachment
SSL	Secure Socket Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
TID	Transport Identification
TLS	Transport Layer Security
TNSX	Transport Name Service in Unix systems
TPI	Transport Protocol Identifier
TS	Transport System
WAN	Wide Area Network

Related publications

The manuals are available as online manuals, see *http://manuals.ts.fujitsu.com*.

openFT for Unix Systems
Installation and Administration
System Administrator Guide

openFT for Windows Systems
Installation and Administration
System Administrator Guide

openFT for Windows Systems
Managed File Transfer in the Open World
User Guide

openFT for Unix Systems and Windows Systems
Program Interface
Programming Manual

openFT for Unix Systems and Windows Systems
openFT-Script Interface
Programming Manual

openFT for BS2000/OSD
Managed File Transfer in the Open World
User Guide

openFT for BS2000/OSD
Installation and Administration
System Administrator Guide

openFT for BS2000/OSD
Program Interface
Programming Manual

openFT for z/OS
Managed File Transfer in the Open World
User Guide

openFT for z/OS
Installation and Administration
System Administrator Guide

CMX
Operation and Administration
User Guide

CMX
Programming Applications
Programming Manual

OSS(SINIX)
OSI Session Service
User's Guide

X/Open CAE Specification
Byte Stream File Transfer (BSFT)
X/Open Document Number XO/CAE/91/400
X/OPEN Company Limited
November 1991

Index

! (BSFT command) 483
%ELEMNAME
  variable 103
%ELEMTYP
  variable 103
%ELEMVERS
  variable 103
%FILENAME 157, 344
  variable 103
%JOBCLASS
  variable 103
%PARTNER 157, 344
  variable 103
%PARTNERAT 157, 344
  variable 103
%RESULT 158, 344
  variable 103
%TEMPFILE 100, 101
%UNIQUE 66
%unique 66
*DELETE (follow-up
  processing) 104
*FTMONITOR 189
/etc/hosts 92
? (BSFT command) 482

A

absolute path name 491
access
  to remote administration
    server 181, 240
access check
  FTAC 53
access control 491
  FTAM attribute 110
access mode 223, 348
access protection 49, 491
  Unix system 98
  Windows 98
access right 491
access rights 161, 214, 348
  display 111
  modify 112
account
  BSFT 460
account number 161
acoustic signal
  activate 461
  deactivate 461
action list 491
activate
  dialog mode 474
  display mode 482
addressing
  partner processor 90
  via Application Entity Title
    (AET) 129
addressing options
  Internet host name 91
  TNS 91
  Transport Name Service 91
ADM partner 91
ADM profile
  create 181
ADM trap server
  outputting the transfer
    admission 300
ADM traps
  setting up a profile on the ADM
    trap server 181, 240
administering
  files (file management) 105
administration 181, 240
admission profile 55, 492
  CSV output format 445
  for collecting monitoring data 189
  privileged 492, 505
  timestamp 248

admission set 51, 492
CSV output format 433
modify 217
privileged 492, 506
Advanced Encryption Standard (AES) 492
AES (Advanced Encryption Standard) 492
AES/RSA 57, 99
AET (Application Entity Title) 492
ANSI code 492
API 373
API (Application Program Interface) 492
append 460
Application Entity Qualifier (AEQ) 129, 131
Application Entity Title (AET) 492
Application Layer 28
Application Process Title (APT) 129
Application Program Interface (API) 492
ASCII 24
ascii 461
asynchronous file transfer
ft command 147
asynchronous request 36, 493
attributes
modifying for remote directory 221
authentication 61, 493
authorization
login 502
LOGON 502
auto-login 453
deactivate 454
automate
sequences 373
automatic restart 39
automation 42
availability
destination file 161, 347

B

background process 493
basic functions 493
behavior on error 375
bell 461
binary 461
binary format 78
binary transfer 81
blank line expansion 78
blanked
file creation password 163, 350
management password 173, 195, 223
transfer admission 154, 172, 191, 194, 202, 203, 211, 212, 222, 257, 258, 341
user password 172
write/read password 155, 192, 212, 258, 342
block-structured 160, 346
BS2000
file types 75
BS2000 computer 350
BS2000 file name
(DVS) syntax 67
(POSIX) syntax 69
BS2000 host 95
BSFT
command mode 453
display status 479
exit 454
start 453
BSFT commands
description 460
overview 457
BSFT messages 484
bye 462

C

calling
BSFT 455
cancel
specification of remote system 462

CCS 85
CCS name
local 155, 201, 342
remote file 156, 201, 342
cd 462
change
local directory 468
order of requests 249
remote directory 462
character repertoire 493
character set 226
for local file 155, 201, 342
for remote file 156, 201, 342
client 493
CLIST procedure, partner
properties 313
close 462
cluster 63
Coded Character Set (CCS) 85
coding
local file 155, 201, 342
remote file 156, 201, 342
collect monitoring data
admission profile 189
Comma Separated Value (CSV) 494
command 143
command execution
remote 41
with postprocessing 44
command syntax 142
commands
file management 139
file transfer 139
instance concept 140
log function 140
long 144
remote execution 41
communication controller 494
compressed transfer 35
compression 98, 150, 335, 494
computer network
open 494, 503
concurrency control 494
connection
establishing with FTP 96
connectivity 494
constraint set 79, 494
contents type 494
convert
to standard admission
profile 234
create 463
FT profile (ftcrep) 174
remote directory 171
sefault admission profile 175
CSV output format 48
admission profile 445
admission set 433
for file attributes 431
general description 145
log record 435
monitoring values 438
operating parameters 442
partner 448
partner properties 300, 313

D

data 495
data communication system 495
data compression 495
data conversion 24
data encoding 495
Data Encryption Standard
(DES) 495
Data Link Layer 28
data protection 495
data security 495
data transfer
POSIX file 75
date 142
date and time of last
modification 109
DDICHK 283
deactivate
auto-login 454
dialog mode 474
display mode 482

default character set 465
default instance 255
define
  separator 479
  write mode 473
define file type
  binary file 461
  BSFT 480
  text file 461
definition
  instance 28
  layer 28
  profile 28
  protocol 27
  service 28
delete 463
  a number of remote files 469
  asynchronous requests 168
  file in a remote system (ftdel) 190
  file in remote system 190
  file in the remote system 463
  FT profiles 196
  log record 61
  remote directory 193
  standard admission profile 196
DENCR 283
DES (Data Encryption
  Standard) 495
DES/RSA 57, 99
description of the output of file
  attributes 260
destination
  ft 150
  ncopy 335
DICHK 283
dir 464
directories
  create 180, 220, 239
  creating remote 171
  delete 180, 220, 239
  deleting remote 193
  display 180, 220, 239
  rename 180, 220, 239
directory 496
display
  access rights 111
  admission set 265
  attributes of a local file 268
  attributes of remote files 256
  FT profiles 306
  log records 272
  operating parameters 300
  partners 311
DNS name 91
document type 79, 496
dynamic partners 90

E

EBCDIC 24, 496
effects
  FT profile 55
emulation 496
encode 465
ENCR 283
encrypted file transfer 99
encryption 57
  old FT versions 58
  request description data 58
  user data 58
enter
  file name 66
entering a file name
  specify 66
entity 496, 500
entries for follow-up processing 143
entries in the command
  sequence 144
error 375
execute shell command 483
exiting
  BSFT 454, 462

F

F-SYSTEM 322
fetch remote file
  BSFT 476
fetching
  multiple files 423

file
  administering 105
  block-structured 160, 346
  delete in remote system 190
  deleting 51
  encrypted transfer 99
  ftamrc 453
  renaming 51
file access rights
  mapping 111
file attributes 496
  CSV- output format 431
  display 180, 220, 238
  modify 180, 220, 238
  modifying 51
  showing 51
file availability
  FTAM attribute 110
file creation password
  blanked 163, 350
file format
  transparent 83
file management 40, 105, 496
  commands 139
  description 105
  FTAM attributes 106
  interplay 51
  local system 106
  remote system 105
file name 142, 171, 190, 193, 211, 221, 226, 257
  specify 56
file name expansion 467
file password 74
  specify 463
file transfer
  commands 139
  encrypted 57
  with postprocessing 505
file transfer request 497
File Transfer, Access and Management 499
file type 148, 226
  BS2000 75
  FTAM 79
  ncopy 333
  Unix system 77
  Windows 77
  z/OS 76
FILE-NAME
  ftshwr output 323
files
  delete 180, 220, 239
  rename 180, 220, 239
filesize 110
filestore 497
firewall processor 497
fixed-length record 497
follow-up processing 42, 44, 104, 497
  %ELEMNAME 103
  %ELEMTYP 103
  %ELEMVERS 103
  %FILENAME 103
  %JOBCLASS 103
  %PARTNER 103
  %PARTNERAT 103
  %RESULT 103
  entries 144
  instance 63
  maximum length 104
  ncopy 343
  overview 103
  variables 103
  with FTAM partners 104
follow-up processing request 497
front-end processor 495
FT administrator 497
ft command 147
FT log record 59
FT profile 52
  delete 196
  display 306
  effects 55
  modify 231
FT request 497, 507

FT system 497
FT trace 497
ft_gzip 419
ft_mget 423
ft_tar 419
FTAC (File Transfer Access Control) 498
FTAC administrator 57, 498
  identify 267
FTAC function 49
FTAC functionality 498
FTAC log record 59
  long output format 286
  reason codes 288
FTAC logging function 498
FTAC messages 415
FTAC transfer admission
  for FTP access 96
ftadm
  protocol prefix 91
FTADM protocol 91
FTAM 32, 499
  file types 79
  kernel group 108
  security group 108
  storage group 108
  virtual filestore 108
FTAM attributes
  kernel group 109
  modify 225
  security group 110
  storage group 109
FTAM catalog 114, 498
FTAM file attributes 498
FTAM partner 32, 79, 499
  addressing 91
  file management 105, 106
  follow-up processing 104
FTAM protocol 499
FTAM standards
  in openFT 32
FTAM-1 79, 496, 498
FTAM-3 79, 496, 498
ftamrc 453
ftcanr 168
ftcredir 171
ftdel 190
ftdeldir 193
ftdelp 196
ftdelp, example 197
ftedit 198
ftexec 200
  messages 204
fthelp 207
ftinfo 208
ftmod 210
ftmoda 217
ftmoddir 221
ftmodf 225
ftmodp 248
ftmodr 249
ftmonitor 251
  calling via a profile 189
ftmsg 254
FTP 29
  inbound access via default FTP 96
FTP partner
  addressing 91
ftseti 255
  messages 255
ftshw 256
ftshwa 265
  example 266
ftshwf 268
ftshwi 270
  messages 271
ftshwl 272
  output 279
ftshwm
  CSV format 438
ftshwo 300
ftshwp 306
  CSV format 145
ftshwptn 311
ftshwr 318
functional standard 499

functionality
of layer 28
future filesize 110, 213

G
gateway 499
gateway processor 499
general string 499
GeneralString 80, 226, 493
get 466
a number of files 470
remote file 475
get file
BSFT 466
glob 467
Gnu zip tools 419
GPL 419
graphical interface
working with 133
GraphicString 80, 226, 493, 499

H
heterogeneous
computer systems 24
link 65
network 27, 499
hidden
user password 194, 222
homogeneous link 65
homogeneous network 27, 500
HOSTS file 500

I
I 323
IA5String 80, 227, 493, 500
identification 500
inbound
file management 51, 500
follow-up processing 50, 500
receive 50, 500
request 500
requests 35, 112
send 50, 500
submission 500
inbound access
FTP 96
inbound mapping
FTAM attributes 115
INBOUND-FILE-MANAGEMENT 266, 267
INBOUND-PROCESSING 266
INBOUND-RECEIVE 266
INBOUND-SEND 266
information
obtaining on standard admission profile 306
on the Internet 19
information on BSFT commands
output 482
information on instances
ftshwi command 270
information on reason codes
output 207
initiator 500
instance 63, 500, 503
definition 28
displaying information on 270
preprocessing, postprocessing follow-up processing 63
preprocessing, postprocessing, follow-up processing 63
selecting 255
setting 63
instance concept
commands 140
instance ID 501
instance identification 61
integrity 501
Internet
information 19
Internet host name
addressing options 91
Internet Protocol (IP) 511, 512
interoperability 501
interplay
file management 51
IPv4 address 92
IPv6 address 92

ISAM file
transferring 160, 346
transferring to a foreign system 83
ISO 8571 32
ISO 8859 24
ISO 8859-1 code table 260
ISO reference model 501
ISO/IEC ISP 10607-3 32
ISO/IEC ISP 10607-6 32
ISO/OSI protocols 32
ISO/OSI reference model 501

J
job 501
transfer 501
job variable
BS2000 47

K
kernel group 79, 109, 498, 501
attributes 109
FTAM 108
key pair set 62

L
LAN (Local Area Network) 501
LAUTH 283
layer
definition 28
lcd 468
legal qualification 163, 349
modify 216
legal qualifications 110
library 501
library element 501
libxml2
license provisions 20
license provisions
libxml2 20
lifetime, request 37
link
heterogeneous 65
homogeneous 65
list
contents of remote directory 464
remote directory 468
loading files in the openFT editor
ftedit 198
Local Area Network (LAN) 501
local system 502
file management 106
locked transfer admissions 376
log function
commands 140
log IDs 279
log records 502
CSV output format 435
output 279
reason codes 207
short output format 279
with postprocessing 279
with preprocessing 279
logging 59
postprocessing 60
preprocessing 60
logging function 502
Logical Unit (LU) 502
login
FTP 96
login admission 52
login authorization 502
LOGON authorization 502
long output format
FTAC log record 286
log record 282
lransport layer 28
ls 468
LU (logical unit) 502

M
mailbox 502
man command 138
managed file transfer 21
management password
blanked 173, 195, 223
manpages 138
mapping of file access rights 111

MAX. ADM LEVELS 178
maximum record length 97
maximum string length 79
maximum-string-length 502
mdelete 469
mdir 469
messages
ftexec 204
ftseti 255
ftshwi 271
openFT 378
mget 470
mls 470
modify
access rights 112
admission set 217
attributes for remote directory 221
file attributes in a remote system 210
FT profile 231
FTAM attributes 225
monitoring data
displaying if monitoring is disabled for partners 292
mput 471

N

ncopy 331
NCP (Network Control Program) 502
network
definition 27
heterogeneous 27, 499
homogeneous 27, 500
Network Control Program (NCP) 502
network description file 503
Network Layer 28
network management 27
networks
openFT support 27
new account number 213, 347
notational conventions 19, 142

O

old FT versions
encryption 58
open 472
open computer network 494
openEdition file 76
syntax 74
openFT add-on products 26
openFT commands 139
openFT Explorer 503
configuration files 136
online help 135
starting 133
openFT for BS2000
partner 503
protocols 503
openFT instances 63
openFT messages 378
openFT partner 31
addressing 91
file management 105
openFT protocol
addressing with 91
openFT protocols 31
openFT-FTAM 503
OPENFTINSTANCE 255
OPENFTLANG 64
operating parameters 504
CSV output format 442
display 300
OSI Reference Model 27
functionality 28
OSI reference model 501
outbound
receive 50, 504
request 504
requests 35, 111
send 50, 504
submission 504
OUTBOUND-RECEIVE 266
OUTBOUND-SEND 266
output
log records 279
remote directories 469, 470

output in CSV format 48, 145
ftshw 431
ftshwa 267, 433
ftshwl 435
ftshwm 438
ftshwo 442
ftshwp 445
ftshwptn 448
output information
on the reason codes 207
output of file attributes
description 260
output working directory
for the remote system 476
outputting
message box on a graphical display 254
system information 208
override 473
owner 504
of FT request 504

P

PAM file
fetching from a foreign system 83
transferring 160, 346
transferring to a foreign system 83
partner 256
CSV output format 448
displaying properties 311
partner address 90, 143
partner list 90
partner name 90, 143, 190, 472
partner processor
addressing 90
partner system 504
partner systems
specified in profile 56
password 155, 173, 191, 195, 212, 223, 258, 341, 504
PDSE member 72
permitted actions 109, 228, 505
physical layer 28
PLAM library
creating 221
deleting 193
PO member 72
port number 505
partner host 92
Portable Open System Interface (POSIX) 505
POSIX (Portable Open System Interface) 505
POSIX file
file format during transfer 75
posix filename (data type) 74
Posix path name (data type) 69
posix pathname (data type) 74
postprocessing 42, 505
ft 151
function 44
instance 63
log record 279
logging 60
ncopy 336
previous FT versions 44
prefix
follow-up processing 56
for file name 56
preprocessing 42, 43, 505
description 100
ft 151
instance 63
log record 279
logging 60
ncopy 336
presentation 505
Presentation Layer 28
presentation selector 505
partner host 93
priority
partners 38
requests 249
priority control 38
PRIV 267
priv 236
private key 505

privileged admission profile 505
privileged admission set 492, 506
privileged profile 236
procedure call
  postprocessing 43, 44
processing
  prohibited (FT profile) 56
  specified 56
product range
  openFT 25
profile 506
  definition 28
  setting up for access to remote administration server 181, 240
  setting up for ADM traps on the ADM trap server 181, 240
profile name 143
program call
  postprocessing 44
  preprocessing 43
program interfaces 46, 373
prompt 474
prompting in procedures 506
protocol 506
  definition 27
PS dataset 71
public key 506
put 475
pwd 476

Q

quit 476
  BSFT 476

R

RAS 506
RAUTH 283
read password
  blanked 155, 192, 212, 258, 342
receive file 506
receive system 506
record 507
record format 227
record length 97, 228, 497, 513
record-by-record transfer 82
recv 476
relative path name 507
remote administration
  access by the remote administration server 181, 240
remote command execution 41
remote directory
  change 462
  creating 171
  deleting 193
  modifying attributes 221
remote system 507
  file management 105
remote transfer admission 154, 171, 190, 194, 202, 211, 222, 257
rename 477
  file in remote system 477
request 507
  asynchronous 36, 493
  file management 105
  lifetime 37
  priority 38
  synchronous 36, 511
request acknowledgment 331
request description data
  encrypting 22
Request for Comments (RFC) 508
request ID 507
request identification 507
request management 507
request number 507
request queue 37, 507
request storage 508
resources 508
responder 508
restart 508
  automatic 39
restart capability
  postprocessing 101
restart point 508

restriction
transfer direction 56
write mode (FT profile) 56
result list 508
RFC (Request for Comments) 508
RFC1006 508
RFC959 29
Rivest-Shamir-Adleman
procedure 508
router 509
RSA procedure 508
RSA/AES 57, 99
RSA/DES 57, 99

S

scope ID 92
SDF procedure, partner
properties 313
SEC-OPTS 283
Secure FTP 509
secure operation 49
security attributes 509
security group 110, 498, 509
attributes 110
FTAM 108
security level 509
sefault admission profile
creating 175
send 478
a number of files 337, 471
file with BSFT 478
file with ft command 147
file with ncopy command 331
send file 509
sending system 509
separator 479
sequence
automate 373
entries in the command 144
sequential file 160, 346
server 509
service 509
definition 28
service class 510
session 510
Session Layer 28
session selector 510
partner host 93
setting an instance 63
ftseti command 255
shell metacharacters 510
shell procedure, partner
properties 313
shell variable
DISPLAY 133
Siemens protocols 31
Simple Network Management
Protocol (SNMP) 510
SN77309 31
SN77312 31
SNA network 510
SNMP 27
SNMP (Simple Network Management
Protocol) 510
source 151, 336
ft 150
ncopy 335
special characters 143, 510
special form (*DELETE) 104
specify
access authorization 481
file name 56
file password 463
file transfer request 65
name of remote system 472
partner processor 90
partner systems in profile 56
prefix for file name 56
prefix for follow-up processing 56
processing 56
syntax rules 97
transfer admission 94
specify account number
BSFT 460
standard admission profile 55
converting to 234
deleting 196
obtaining information 306

standard admission set 265, 510
standard error output (stderr) 510
standard input (stdin) 510
standard output
  ftshw 260
standard output (stdout) 511
starting
  BSFT 453
status
  BSFT 479
status message
  ncopy 332
std instance 255
stderr 510
stdin 510
stdout 511
storage group 109, 498, 511
  attribute 109
  FTAM 108
string 511
string significance 79, 511
symbolic link 182
synchronous file transfer
  ncopy command 331
synchronous request 36, 511
syntax
  BS2000 file name (DVS) 67
  BS2000 file name (POSIX) 69
  Unix system file name 70
  Windows file name 70
  z/OS file name 71
syntax rules
  specify 97
system 511
  local 502, 511
  remote 507, 511

T

T-selector 512
tabulator expansion 78
TCP/IP 511, 512
text format 77
  data conversion 24
text transfer 80
timestamp
  showing for admission profile 310
  updating on admission profile 248
TNS
  addressing options 91
TNS (Transport Name Service) 512
tool command library 419
transfer
  encrypted 99
  in binary format 81
  in text format 80
  in user format 81
  record-by-record 82
  transparent format 83
transfer admission 143, 154, 511
  blanked 154, 172, 191, 194, 202, 203, 211, 212, 222, 257, 258, 341
  file transfer request 55
  FTAC 52
  locked 376
  outputting (ADM trap server) 300
  specify 94
transfer direction
  restriction 56
transfer file
  DVS file 75
  file name syntax 75
  library element 75
  PLAM library 75
  POSIX file 75
transfer identification 511
transfer unit 512
Transmission Control Protocol (TCP) 511, 512
transparent file format 83
transparent format
  transfer 83
transport connection 512
transport layer 512
Transport Name Service
  addressing options 91

Transport Name Service (TNS) 512
transport protocol 29, 512
transport selector 512
  partner host 92
transport system 29, 31, 512
type 480
types
  follow-up processing 103

U
umlauts
  data conversion 24
UNC names 70
Unicode 24
universal-class-number 513
Unix system
  access protection 98
  file name, syntax 70
  file types 77
UNIX(TM) 513
user 481
user data
  encryption 58
user format 79
  transfer 81
user ID 143
user password
  blanked 172
  hidden 194, 222
using disabled basic functions 178

V
variable-length record 513
variables
  follow-up processing 103
  follow-up processing (ft) 157
  follow-up processing (ncopy) 343
verbose 482
virtual filestore 32, 108, 513
  FTAM 108
VisibleString 80, 227, 493
visibleString 513
VSAM file 73

W
WAN (Wide Area Network) 514
what if ... 375
Wide Area Network (WAN) 514
wildcards 337
  ft_mget 423
Windows
  access protection 98
  file types 77
Windows file name
  syntax 70
Windows procedure, partner
  properties 313
wirte mode
  restriction 56
write mode 149
  ncopy 334
write password
  blanked 155, 192, 212, 258, 342

X
X terminal 514
X Window interface 133

Z
z/OS
  file name, syntax 71
  file type 76
z/OS UNIX System Services 74
zip compression 150, 335
zip tools 419